Wenn Sie bisher glaubten, Mikroprozessoren und Mikrocomputer seien erst nach einem Informatik-Studium oder nach einigen Jahren EDV-Praxis zu verstehen, dann werden Sie in unserer neuen Serie gründlich eines anderen belehrt. Und zwar nicht etwa auf die typisch mathematisch-trockene Art, sondern auf eine Weise, die Ihnen sicherlich auch ein wenig Spaß bereitet. Was Sie mitbringen müssen? Nur ein wenig Sinn für Humor.

Hans Werner

Mikrocomputer

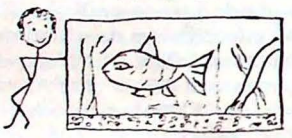
Eine kleine Einführung

1. Teil

Im 1. Teil...

wird ein sehr einfaches Computermodell entworfen. An dem Modell werden Schaltungsprobleme und kleine Programme erläutert. Die Software beschränkt sich vorläufig auf 8 Befehle.

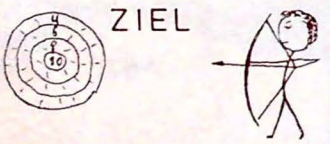
INHALT



Der 2. Teil...

befaßt sich mit realen Mikrocomputern, allerdings in vereinfachter Form. Der Befehlsvorrat wird stark erweitert, so daß alle wichtigen Adressierarten möglich sind.

ZIEL



ist es,

1. einfache Fachliteratur lesen und auswerten zu können,
2. den Begriff MIKROCOMPUTER vom Mythos des Alleskönnens zu befreien.

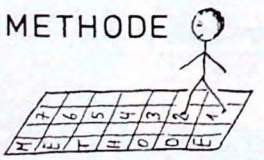
Ing. (grad.) Hans Werner ist 34 Jahre alt und in Ostpreußen geboren. An der Fachhochschule Furtwangen studierte er Elektronik und Regelungstechnik. Dann war er im Schaltanlagenbau bei der Energieversorgung Weser-Ems und später in Konstruktion und Prüffeld der Fa. Carl Schenck in Darmstadt tätig; bei dieser Firma ist er heute Gruppenleiter für den Bereich Technische Information. Seine Hobbys sind ein wenig ausgefallen: Bau von Buddelschiffen und Kleinstflugmodellen.

...Systeme mit Mikrocomputern entwickeln zu können. Das ist nur nach intensivem Studium der Firmen-Fachliteratur möglich.

ZIELGRUPPE

Elektriker und andere Interessierte. Die Vorkenntnisse beschränken sich auf das binäre Zahlensystem und auf elementare Kombinationsschaltungen.

METHODE



Kleine Schritte sollen rationell und ohne viel Aufwand zum Ziel führen. Die einzelnen Sachgebiete sind ineinander verzahnt und nicht wie ein Lexikon gegliedert. Kritiker bezeichnen diese Methode auch als konzeptlos.

Alle Erklärungen fangen mit der Hardware an. Dieser Weg kommt ohne abstrakte Denkmodelle aus und erscheint für die Zielgruppe sinnvoll.

KONTROLLE

Tests:

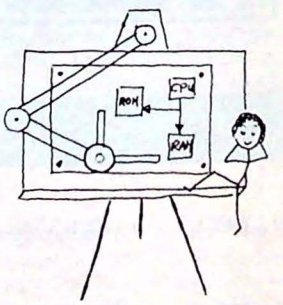
a) mit Auswahlantworten

b) mit Lückentext

c) mit kleinen Aufgaben

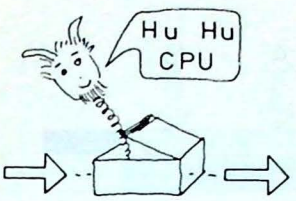
d) mit Rätseln, die an den Lehrstoff gebunden sind. Sie können übergangen werden.

Einleitung

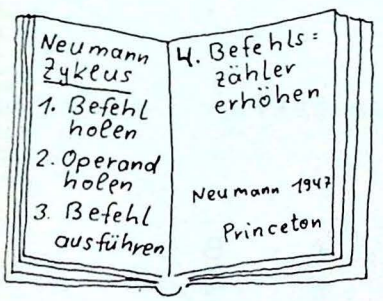


Ein Mikroprozessor ist die Zentraleinheit eines Mikrocomputers und alleine nicht funktionsfähig. Deshalb wird vorläufig auf eine genaue Trennung und Erklärung von Begriffen wie CPU und Interface verzichtet.

Vielmehr wird aus einzelnen Bausteinen ein sehr einfacher, aber kompletter Mikrocomputer entworfen, der natürlich nur einen pädagogischen Sinn hat. Selbstverständlich kann nicht



jede Kleinigkeit erklärt werden. Deshalb werden die einzelnen Funktionsgruppen durch sogenannte schwarze Kästen dargestellt und nur das Verhalten der Ein- und Ausgänge genau festgelegt.



Die Entwicklung von Computern ging in 3 Abschnitten vor sich:

1800

entstand der erste programmierbare Rechenautomat von Babbage.

1910

führte Konrad Zuse das Dualsystem für Computer ein und baute den ersten Dualrechner.

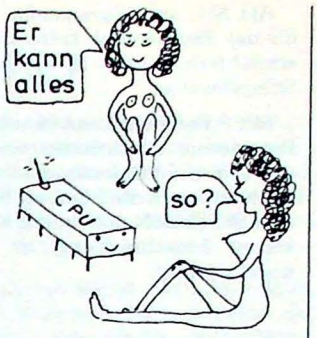
1947

legte John von Neumann den prinzipiellen Ablauf der Programmsteuerung fest. Der „Von-Neumann-Zyklus" ist noch heute uneingeschränkt gültig. Seitdem haben alle Computer folgende Punkte gemeinsam:

1. Der Aufbau des Computers ist unabhängig von dem zu bearbeitenden Problem.

2. Zerlegung in die logischen Gruppen: Rechenwerk, Speicherwerk, Steuerteil, Ein-Ausgabewerke.

3. Es wird mit Dualzahlen gearbeitet.

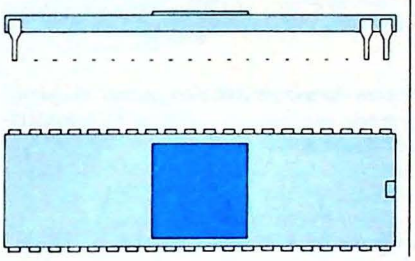


Mikrocomputer können alles!

Das ist natürlich stark übertrieben, obwohl grundsätzlich jedes Problem der Meß-, Steuer- und Regelungstechnik gelöst werden kann.

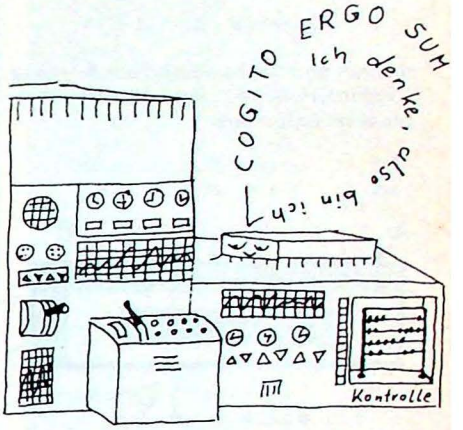
Aber schon bei schnellen Regelkreisen versagt der viel zu langsame Mikrocomputer.

Die einfache Ausführung eines Mikrocomputers besteht aus ein bis zwei integrierten Schaltkreisen.



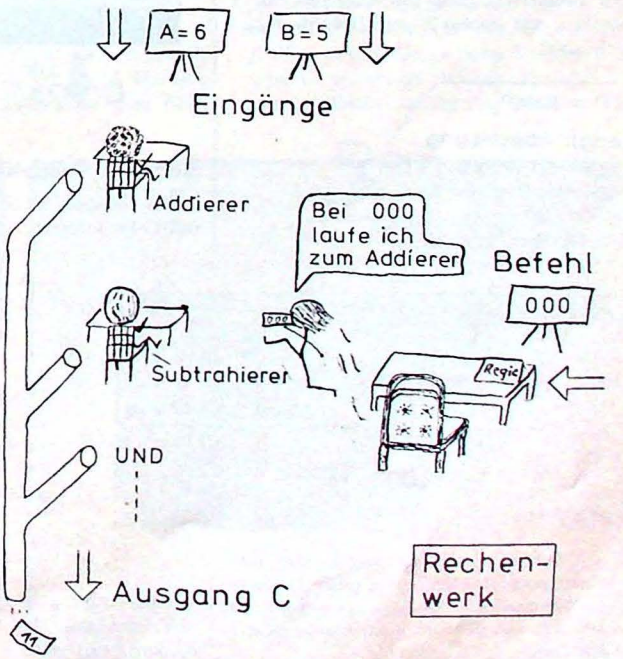
Das Bild zeigt in etwa die Größe eines solchen Schaltkreises. Seit Zuse ist die räumliche Ausdehnung eines Computers etwa um den Faktor 1 000 000 und mehr zurückgegangen.

Das Rechenwerk



Eine wichtige Komponente jedes Computers ist das Rechenwerk. Es hat 2 Dateneingänge A und B, an denen z. B. die Werte 5 und 6 stehen. Die Eingangsdaten können addiert, subtrahiert und logisch verknüpft werden.

Die Art der Operation hängt vom Befehlseingang ab:

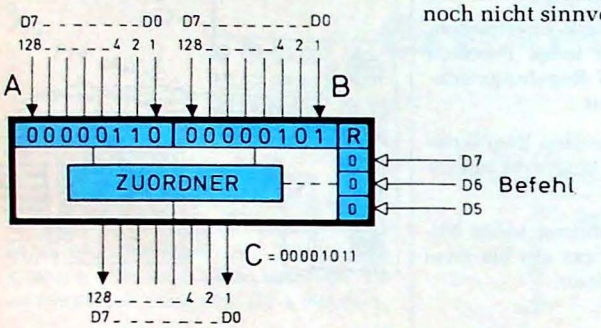


Der Befehlseingang bestimmt das Verhalten des Rechenwerkes.

Der Befehl 000 bedeutet z. B. ADDIEREN. Das Befehlsmännchen läuft zum Addierer, der eifrig

$$A + B = 6 + 5 = 11$$

rechnet und das Ergebnis zum Ausgang C befördert. Bei 001 wird der Subtrahierer verständigt usw.



Das Rechenwerk besteht aus verschiedenen Zuordnern:

1. für arithmetische Operationen (+ und –)[1]
2. für logische Operationen (UND, ODER)
3. für Transportbefehle (Speichern, Laden usw.)

Die Dateneingänge A, B und Ergebnisausgang C haben 8 Binärstellen $2^0 \ldots 2^7$ (D0=1, D1=2, D2=4...D7=128). Der Befehlseingang hat dagegen nur 3 Stellen, mit denen $2^3 = 8$ Befehle möglich sind.

Befehl	Kürzel	Erklärung	
000	AD	Addieren	C = A + B
001	SU	Subtrahieren	C = A – B
010	UN	logisch UND	C = A ∧ B
011	OD	logisch ODER	C = A ∨ B
100	LA	Laden	C = B
101	SP	Speichern	C = A
110	JP	Sprung	C = A
111	ST	Stop	

[1]) Einige Mikrocomputer haben keinen Subtraktionsbefehl. Bei ihnen muß die Subtraktion auf die Addition zurückgeführt werden. Das geht im Binärsystem relativ einfach.

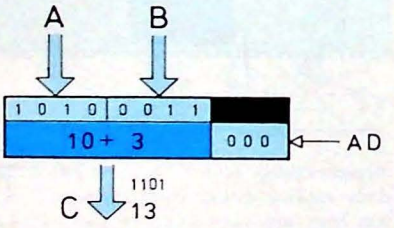
AD, SU...sind Kurznamen. Sie haben für das Rechenwerk keine Bedeutung, erleichtern aber in Zukunft die Schreibweise.

Mit 8 Befehlen können schon kleine Programme geschrieben werden, obwohl für die Minimalausstattung eines Rechners noch die 2 Befehle NEGIEREN und SCHIEBEN notwendig sind. Diese kleine Einschränkung ist vorläufig nicht wichtig.

Die Befehle LA, SP, JP und ST sind bei der Darstellung des Rechenwerkes noch nicht sinnvoll zu erklären und erscheinen erst im weiteren Verlauf als notwendig.

Die Operationszeit des Rechenwerkes soll vernachlässigbar klein sein, d. h. die Daten A, B und der Befehl müssen gleichzeitig an den Eingängen stehen. Anders ausgedrückt:

Das Ergebnis ist erst dann sinnvoll, wenn die Signale auf allen Eingangsleitungen stabil sind.



Addieren, Subtrahieren

Als Beispiel ist eine Addition (Befehl 000) eingetragen.

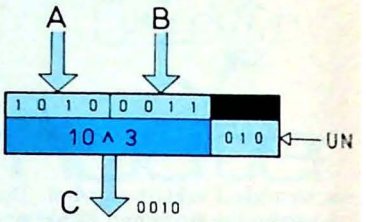
A	10	1010
B	+3	+0011
C	13	1101

Das Rechenwerk-Kästchen ist nur für 4 Bit gezeichnet. In Wirklichkeit sind es 8, so daß die größte darstellbare Zahl $1 + 2 + 4 + 8 + 16 + 32 + 64 + 128 = 255$ beträgt.

Bei Zahlen größer als 255 ergibt sich ein Übertrag, der vorläufig nicht berücksichtigt wird.

Liegt 001 am Befehlseingang, wird subtrahiert.

A	10	1010
B	–3	–0011
C	7	0111



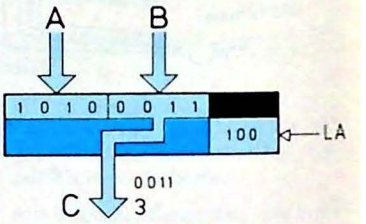
UND, ODER

1010 UND 0011

A	1010
B	0011
C	0010

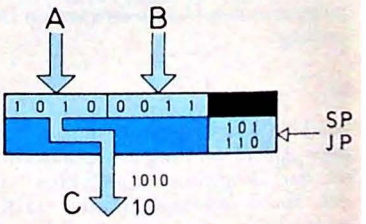
1010 ODER 0011

A	1010
B	0011
C	1011



Laden

Der Eingang B wird durchgeschaltet. A hat keinen Einfluß.



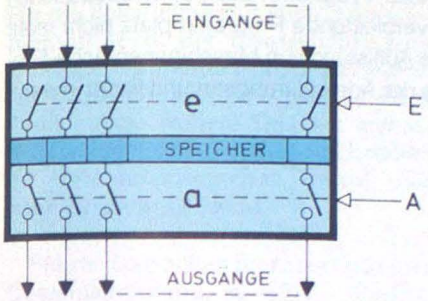
Speichern, Sprung

Der Eingang A wird durchgeschaltet. B hat keinen Einfluß.

Stop

Der Befehl ST hat keinen Einfluß auf das Rechenwerk und wird später erläutert.

Speicher



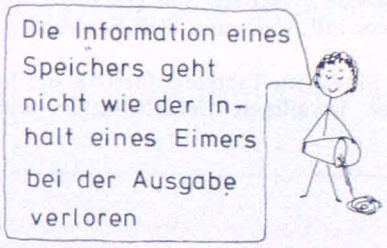
Ein weiterer wichtiger Baustein ist der Speicher, der normalerweise aus mehreren Speicherzellen besteht. Jede Speicherzelle kann entweder den Zustand 0 oder 1 annehmen. Das kann im primitivsten Fall durch Relais mit Selbsthaltung oder durch Flipflops geschehen.

Ein Speicher kann Daten in binärer Form festhalten.

Für Speicher, die nur kurzzeitig Daten festhalten sollen, hat sich auch der Name *Register* eingebürgert.

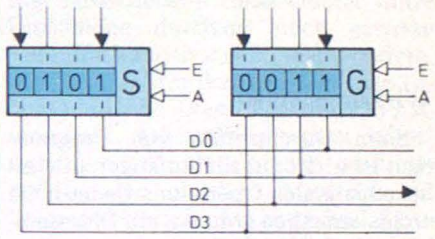
Üblicherweise lassen sich die Speicherzellen durch Schalter von den Ein- und Ausgängen trennen. Gesteuert werden sie mit den beiden Steuereingängen E und A.

E = Eingabe		A = Ausgabe		
E	*A*	*e*	*a*	*Funktion*
0	0	geöffnet	geöffnet	keine
0	1	geöffnet	geschl.	Ausgabe
1	0	geschl.	geöffnet	Eingabe



Wenn E und A an 0 liegen, sind alle Schalter geöffnet. Liegt A an 1, kann der Inhalt aller Speicherzellen an den Ausgängen abgefragt werden. Liegt E an 1, werden die Eingangswerte in den Speicher eingeschrieben.

Die Speicherausgänge können die Zustände 0,1 oder hochohmig (Schalter offen) annehmen. Deshalb werden solche Ausgänge auch als Tri-State-Ausgänge bezeichnet.



Aufgabe

2 Speicher arbeiten auf gemeinsame Ausgangsleitungen.
Darf an beiden Steuereingängen A gleichzeitig 1 stehen?

(Fortsetzung folgt)

Zilog setzt auf 16-bit-Mikroprozessoren

Mit dem Z8000, der im Herbst in Deutschland verfügbar sein wird, will *Zilog* an das obere Ende des Kleinrechnermarktes vorstoßen, gab H. P. Blomeyer, Leiter des Geschäftsbereichs Mikroprozessoren der Fa. *Kontron Elektronik*, anläßlich einer Pressekonferenz bekannt.

Die Fa. Zilog, die sich auf dem heißumkämpften Mikroprozessormarkt einen entscheidenden Marktanteil erkämpft hat, bringt hier einen 16-bit-Mikroprozessor mit einer sehr komfortablen Architektur auf den Markt (Befehlszykluszeit: 750 ns). Die CPU Z8000 kann bis zu 8 MByte Speicher direkt adressieren. Adressierbare Elemente sind Bits, BCD-Digits, Bytes, Worte (16 bit), Doppelworte sowie Folgen von Worten. Der Datendurchsatz liegt im Schnitt höher als bei der PDP11/45.

Mit dem Z8, einem Einchip-8-bit-Mikrocomputer mit 2 KByte Programmspeicher (ROM), 128 Byte RAM, 32 programmierbaren Eingabe-/Ausgabeleitungen, serieller Schnittstelle (UART), zwei Zählern/Zeitgebern sowie einer sehr ausgereiften CPU, kommt ein Mikrocomputer auf den markt, der alle bisher auf dem Markt befindlichen Einchip-Mikrocomputer übertreffen soll.

Der Präsident der Firma, Dr. F. Faggin, gab einige Daten über die Firma Zilog bekannt: Seit der Firmengründung 1974 ist der Betrieb auf 300 Mitarbeiter angewachsen. Ende 1978 wird man, wenn alles nach Plan läuft, 1000 Mitarbeiter haben. Der Umsatz wird von 1,5 Mio. US-$ (1976) auf 25 Mio. US-$ (1978) ansteigen. Mehr als 50% der Firma werden von der *Exxon Enterprises* gehalten. Erfreuliche (oder erschreckende?) Ausblicke wurden aufgezeigt: Man wird bis 1985 in der Lage sein, 45000 Gatter oder 1 Mio. Bit Speicher auf einem Chip zu integrieren. Der Einfluß solcher Technologie auf unseren Lebensstil und auf unser Geschäftsleben wird enorm sein. -yr

Platinen zu der Reihe „Praxis mit Mikroprozessoren"

In FUNKSCHAU 1978, Heft 10, wurde die Reihe „Praxis mit Mikroprozessoren" von Horst Pelka beendet. Das darin für den Selbstbau beschriebene 8080-System war ursprünglich in Fädeltechnik auf Lochrasterplatten aufgebaut worden; jetzt sind aber auch Platinen dafür erhältlich, und zwar sowohl unbestückte als auch bestückte und geprüfte.

Über unsere Beschreibungen hinaus stehen auch Platinen und Moduln für hexadezimale Eingabe und Anzeige von Adressen und Daten, ein EPROM-Programmiermodul, ein Netzteilmodul und anderes Nützliches zur Verfügung.

Die Moduln sind auf Europakarten untergebracht und lassen sich in einen passenden Einschubrahmen mit Griffen, Führungsschienen und 72 E/A-Leitungen einfügen. Nähere Informationen sind bei Hilmar Frehsdorf, Rotdornweg 42, 2085 Quickborn zu erhalten.

Mikrocomputer-Ausstellung

Parallel zur Electronica, nämlich vom 9. bis 15. November 1978, findet im Pschorr-Keller neben dem Münchner Messegelände die „Industrie und Personal Computer Expo '78", kurz IPE'78, statt. Etwa 25 Firmen aus Europa und den USA stellen Mikrocomputer nebst Peripherie aus. Die Veranstalter rechnen mit etwa 5...10% der Electronica-Besucher. Der Franzis-Verlag wird ebenfalls auf der IPE vertreten sein.

Wer einen KIM-1 und einen ASCII-Fernschreiber, einen Drucker oder ein Terminal besitzt, kann das hier vorgestellte Disassembler-Programm auch ohne zusätzlichen Speicher bereits einsetzen, wenn auch der verbleibende Programmplatz nicht mehr allzu groß ist. Der Disassembler erzeugt eine Auflistung von Maschinensprache-Programmen in mnemonischer Form mit Angabe der Adressierungsart und findet in weniger als 0,8 kByte Platz.

6502-Disassembler

Wirkungsweise

Beim Durchprüfen von Programmen ist nicht nur für Anfänger, die mit hexadezimalen Operations-Codes noch nichts anfangen können, ein Disassembler sehr hilfreich. Die mnemonischen Befehlsbezeichnungen, die er für jeden Operationscode liefert, sind normalerweise sofort verständlich, z.B. bedeutet LDA soviel wie „Load Accu".

Eine weitere Hilfestellung bietet der hier beschriebene Disassembler bei der Angabe der Adressierungsart. Ist sie z.B. indirekt, so wird das Argument in Klammern gesetzt, und bei indizierter Adressierung wird ein Komma und das jeweilige Indexregister hinter das Argument gesetzt. Die bei Branch-Befehlen verwendete relative Adressierung wird in absolute, vierstellige Adressen umgerechnet, um sofort das Sprungziel deutlich zu machen.

Falls der Disassembler auf einen Operationscode stößt, der illegal ist (z.B. FF), hält er automatisch an, springt zum KIM-Monitorprogramm an die Adresse 1929 und druckt die Adresse FFFF als Fehlermeldung aus. Normalerweise erfolgt die Auflistung des Programms, wenn keine solche Fehlermeldung vorkommt, von der Band-Startadresse bis zur Band-Endadresse. Diese Adressen müssen in den Zellen 17F5...17F8 gespeichert sein. *Bild 1* zeigt ein grobes Flußdiagramm des Programmablaufes.

Anzumerken wäre noch, daß das Disassembler-Programm eine Reihe von Unterprogrammen des im KIM-ROM gespeicherten Monitors benutzt. Dies muß beachtet werden, wenn das hier beschriebene Programm in anderen 6502-Systemen verwendet werden soll.

Die Decodierung

Das „Hauptproblem" eines Disassemblers ist natürlich, die hexadezimalen Maschinenbefehle, die Operationscodes also, zu decodieren. Um Speicherplatz zu sparen, geschieht das hier aber nicht für jeden einzelnen Operationscode; vielmehr werden die Codes gezielt über Tabellen zunächst maskiert, so daß nur die gerade interessierenden Bits übrigbleiben, und diese werden schließlich zum Teil über eine zweite Tabelle invertiert. (Das geschieht übrigens mit AND- und EOR-Funktionen.)

Eine dritte Tabelle enthält die sich bei der jeweiligen Befehlsgruppe ergebende Länge (1, 2 oder 3 Bytes), und eine vierte gibt an, wo das Programm nach der Decodierung hinspringen soll, um die so ermittelte Adressierungsart zu speichern. Alle diese vier Tabellen werden parallel durchlaufen, wobei das X-Register zur Indizierung verwendet wird. So funktioniert zunächst einmal die Feststellung der Adressierungsart.

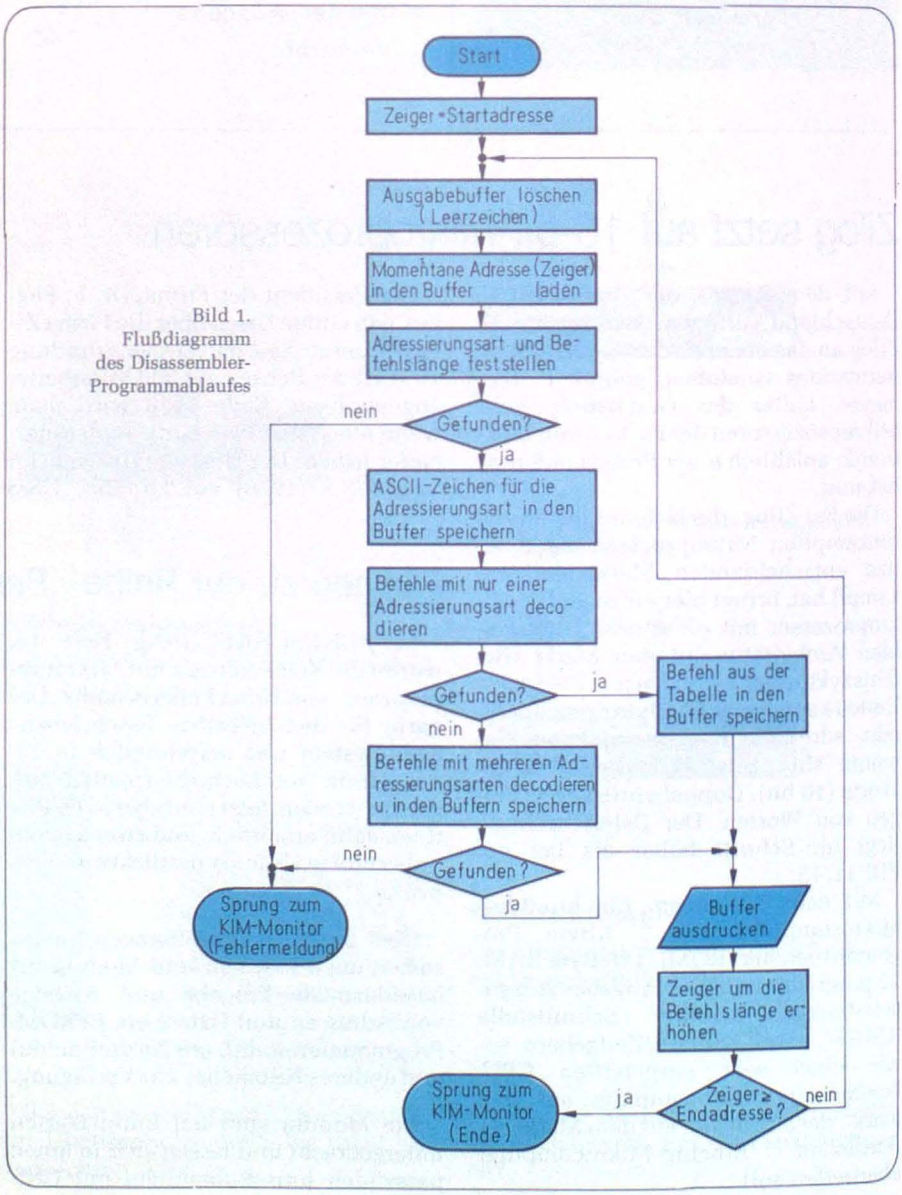


Bild 1. Flußdiagramm des Disassembler-Programmablaufes

Auch die mnemonischen Codes werden mit Hilfe von Tabellen gewonnen. Für die Befehle mit nur einer Adressierungsart sucht der Disassembler eine Tabelle ab, die direkt die hexadezimalen Codes enthält. Drei parallel dazu durchlaufene weitere Tabellen enthalten die zugehörigen ASCII-Buchstaben für den mnemonischen Befehl, der dann ausgedruckt wird.

Führte das Suchen hier aber zu keiner Übereinstimmung, so wird – ähnlich wie bei der Feststellung der Adressierungsart – der Befehl durch Maskieren und Invertieren einzelner Bits decodiert. Dazu dienen vier weitere Tabellen, die die AND- und EOR-Argumente sowie die drei ASCII-Zeichen für den jeweiligen Befehl enthalten.

Ausgabe der Befehle

Der so decodierte Befehl steht nun zunächst einmal in einem Ausgabe-Buffer zur Verfügung (Adressenbereich 01DO...01DF). Normalerweise wird nun der gesamte Bufferinhalt seriell mit Hilfe des KIM-Monitor-Unterprogramms OUTCH (JSR 1EA0) und einer kleinen Schleife seriell zum Fernschreiber befördert; will man dagegen einen einfachen Drucker ansteuern, erweist sich dieser Zeichenbuffer als besonders nützlich, weil solche Drucker meist von rechts nach links ausgeben.

Der Buffer enthält das in der Ausgabezeile ganz links stehende ASCII-Zeichen an der Adresse 01D0, das ganz rechts stehende an der Adresse 01DF.

Um die Befehle auf einem kleinen Drucker auszugeben, wird ein besonderes Programm benötigt (0112), das individuell den Bedürfnissen des Druckers angepaßt werden muß und deshalb hier nicht aufgelistet ist; es ist lediglich statt des Fernschreiber-Ausgabeprogramms in den Programmablauf einzufügen.

Es sei nicht verschwiegen, daß der hier beschriebene Disassembler unter Umständen durchaus noch gewisse Vereinfachungen zuläßt, insbesondere, was die Struktur der Decodierungs-Tabellen angeht. Diese Tabellen sind so aufgebaut, daß auch zwischen Adressierungsarten unterschieden wird, die zufällig den gleichen Ausdruck zur Folge haben, z.B. Absolute und Zero-

```
DISASSEMBLER
BRANCH-ADR.ABS.

START: 0000
0000 LDF 17F5
0003 STA   FA
0005 LDA 17F6
0008 STA   FB
000A LDX #$10
000C LDA #$20
000E STA 01CF,X
0011 DEX
0012 BNE 000C
0014 LDA   FB
0016 JSR 013D
0019 STY 01D0
001C STA 01D1
001F LDA   FA
0021 JSR 013D
0024 STY 01D2
0027 STA 01D3
002A NOP
002B NOP
002C NOP
002D LDX #$00
002F LDY #$00
0031 LDA  (FA),Y
0033 AND 0180,X
0036 EOR 01C0,X
0039 BEQ 0048
003B INX
003C CPX #$10
003E BNE 002F
0040 LDA #$3F
0042 STA 01DF
0045 JMP 1929
0048 LDA 03D8,X
004B STA   F4
004D LDA #$00
004F STA   F0
0051 LDA 03E8,X
0054 STA   EF
0056 LDX   F4
0058 DEX
0059 BEQ 0078
005B DEX
005C BEQ 006B
005E LDY #$02
0060 LDA  (FA),Y
0062 JSR 013D
0065 STY 01D9
0068 STA 01DA
006B LDY #$01
006D LDA  (FA),Y
006F JSR 013D
0072 STY 01DB
0075 STA 01DC
0078 JMP(00EF)
007B LDA #$23
007D STA 01D9
0080 LDA #$24
0082 BNE 0086
0084 LDA #$41
0086 BNE 00A7
0088 LDA #$29
008A STA 01DD
008D LDA #$2C
008F STA 01DE
0092 LDA #$59
0094 BNE 00A2
0096 LDA #$2C
0098 STA 01DD
009B LDA #$58
009D STA 01DE
00A0 LDA #$29
00A2 STA 01DF
00A5 LDA #$28
00A7 STA 01DA
00AA BNE 00C3
00AC LDA #$58
00AE BNE 00B2
00B0 LDA #$59
00B2 STA 01DE
00B5 LDA #$2C
00B7 BNE 00C0
00B9 LDA #$28
00BB STA 01D8
00BE LDA #$29
00C0 STA 01DD
00C3 NOP
00C4 NOP
00C5 NOP
00C6 LDY #$00
00C8 LDA  (FA),Y
00CA LDX #$20
00CC CMP 02DF,X
00CF BEQ 00D7
00D1 DEX
00D2 BNE 00CC
00D4 JMP 0185
00D7 JMP 0100
00DA JMP 0157
0100 LDA 02FF,X
0103 STA 01D5
0106 LDA 031F,X
0109 STA 01D6
010C LDA 033F,X
010F STA 01D7
0112 JSR 1E2F
0115 LDX #$00
0117 LDA 01D6,X
011A JSR 1EA0
011D INX
011E CPX #$10
0120 BNE 0117
0122 JSR 1F63
0125 LDA   FA
0127 CMP 17F7
012A BNE 0133
012C LDA   FB
012E CMP 17F8
0131 BEQ 013A
0133 DEC   F4
0135 BNE 0122
0137 JMP 000A
013A JMP 1C4F
013D TAX
013E LSR  A
013F LSR  A
0140 LSR  A
0141 LSR  A
0142 JSR 014B
0145 TAY
0146 TXA
0147 JSR 014B
014A RTS
014B AND #$0F
014D CMP #$0A
014F CLC
0150 BMI 0154
0152 ADC #$07
0154 ADC #$30
0156 RTS
0157 LDA   FB
0159 STA   F9
015B LDY #$01
015D LDA  (FA),Y
015F BPL 0163
0161 DEC   F9
0163 CLC
0164 ADC #$02
0166 BCC 016A
0168 INC   F9
016A CLC
016B ADC   FA
016D BCC 0171
016F INC   F9
0171 JSR 013D
0174 STA 01DC
0177 STY 01DB
017A LDA   F9
017C JSR 013D
017F STY 01D9
0182 JMP 00A7
0185 LDX #$18
0187 LDY #$00
0189 LDA  (FA),Y
018B AND 03A7,X
018E EOR 03BF,X
0191 BEQ 0199
0193 DEX
0194 BNE 0189
0196 JMP 1929
0199 LDA 035F,X
019C STA 01D5
019F LDA 0377,X
01A2 STA 01D6
01A5 LDA 038F,X
01A8 STA 01D7
01AB JMP 0112
```

Bild 2. Hier hat sich der Disassembler selbst aufgelistet

Page (der Unterschied besteht hier nur in der Länge des Arguments). Dies ist zwar nicht unbedingt erforderlich, vereinfacht jedoch spätere Erweiterungen oder die Umstellung auf andere Darstellungsarten erheblich.

Belegter Speicherplatz

Das Disassembler-Programm (*Bild 2*) wurde so ausgelegt, daß es in den im KIM-1 in der Grundstufe vorhandenen Speicherplatz paßt. Dabei wurde darauf geachtet, daß auch solche Anwenderprogramme disassembliert werden können, die an der meist üblichen Startadresse 0200 beginnen. Die Länge dieser Anwenderprogramme ist hier allerdings bis zur Adresse 02DF begrenzt, denn dort beginnen einige für den Disassembler erforderliche Code-Tabellen. *Bild 3* zeigt die hexadezimale Programmauflistung mit diesen Tabellen.

Eine Verlegung des Disassemblers in einen anderen Speicherbereich ist bei erweiterten Systemen durchaus möglich. Dabei ist zu beachten, daß außer den Adressen auch der LDA-00-Befehl, der die für einen indirekten Sprung während der Adressierart-Decodierung nötige Zieladresse als höherwertige Byte in die Zelle 00F0 speichtert (Page der Zieladresse), geändert werden muß.

Herwig Feichtinger

PET 2001 mit Bandzählwerk

Sollten Sie auch einen PET besitzen und auf jede Ihrer Kassetten immer mehrere Programme nacheinander laden, dann ist es ganz praktisch, ein Bandzählwerk in den Kassettenrecorder einzubauen. In unserer Rubrik P+H erfahren Sie, wie man das am besten macht; die Anzeige hat im PET bequem auf der Vorderfront über dem Recorder Platz.

```
0000 AD F5 17 85    00B0 A9 59 8D DE    0180 D9 01 4C A7    033C 58 59 45 45
0004 FA AD F6 17    00B4 01 A9 2C D0    0184 00 A2 18 A0    0340 4B 43 53 51
0008 85 FB A2 10    00B8 07 A9 28 8D    0188 00 B1 FA 3D    0344 49 45 4C 43
000C A9 20 9D CF    00BC D8 01 A9 29    018C A7 03 5D BF    0348 53 43 44 49
0010 01 CA D0 F8    00C0 8D DD 01 EA    0190 03 F0 06 CA    034C 56 58 59 56
0014 A5 FB 20 3D    00C4 EA EA A0 00    0194 D0 F3 4C 29    0350 59 50 41 50
0018 01 8C D0 01    00C8 B1 FA A2 20    0198 19 BD 5F 03    0354 41 50 49 53
001C 8D D1 01 A5    00CC DD DF 02 F0    019C 8D D5 01 BD    0358 58 59 58 41
0020 FA 20 3D 01    00D0 06 CA D0 F8    01A0 77 03 9D D6    035C 53 41 43 44
0024 8C D2 01 8D    00D4 4C 85 01 4C    01A4 01 BD 8F 03    0360 45 53 53 52
0028 D3 01 EA EA    00D8 00 01 4C 57    01A8 8D D7 01 4C    0364 4F 4C 4C 4C
002C EA A2 00 A0    00DC 01             01AC 12 01 00 00    0368 4C 43 41 41
0030 00 B1 FA 3D    0100 BD FF 02 8D    01B0 FF FF FF DF    036C 41 4A 53 53
0034 B0 01 5D C0    0104 D5 01 BD 1F    01B4 9F 1F 8D 1F    0370 52 49 44 43
0038 01 F0 0D E8    0108 03 8D D6 01    01B8 1F 1C 1C 1C    0374 43 42 4A 53
003C E0 10 D0 EF    010C BD 3F 03 8D    01BC 1C 19 05 14    0378 4F 54 42 4F
0040 A9 3F 8D DF    0110 D7 01 20 2F    01C0 20 6C BE 96    037C 52 53 44 44
0044 01 4C 29 19    0114 1E A2 00 BD    01C4 0A 10 80 11    0380 44 4D 53 4E
0048 BD D8 03 85    0118 D0 01 20 A0    01C8 01 04 0C 14    0384 44 4D 54 54
004C F4 A9 00 85    011C 1E E8 E0 10    01CC 1C 19 00 00    0388 4F 4E 45 50
0050 F0 BD E8 03    0120 D0 F5 20 63    02E0 00 90 B0 F0    038C 50 49 53 45
0054 85 EF A6 F4    0124 1F A5 FA CD    02E4 30 D0 10 50    0390 52 41 43 4C
0058 CA F0 1D CA    0128 F7 17 D0 07    02E8 70 18 D8 58    0394 41 52 59 58
005C F0 0D A0 02    012C A5 FB CD F8    02EC B8 CA 88 E8    0398 41 50 4C 44
0060 B1 FA 20 3D    0130 17 F0 07 C6    02F0 C8 EA 48 08    039C 43 50 59 58
0064 01 8C D9 01    0134 F4 D0 EB 4C    02F4 68 28 40 60    03A0 52 43 43 58
0068 8D DA 01 A0    0138 0A 00 4C 4F    02F8 AA A8 BA 8A    03A4 59 54 52 49
006C 01 B1 FA 20    013C 1C AA 4A 4A    02FC 9A 98 38 F8    03A8 E3 E3 E3 E3
0070 3D 01 8C DB    0140 4A 4A 20 4B    0300 42 42 42 42    03AC E3 E3 E3 E3
0074 01 8D DC 01    0144 01 A8 8A 20    0304 42 42 42 42    03B0 E3 E3 E3 E3
0078 6C EF 00 A9    0148 4B 01 60 29                        03B4 E3 DF E7 E7
007C 23 8D D9 01    014C 0F C9 0A 18    0308 42 43 43 43    03B8 E7 E7 E7 F3
0080 A9 24 D0 02    0150 30 02 69 07    030C 43 44 44 49    03BC F3 F7 FF FF
0084 A9 41 D0 1F    0154 69 30 60 A5    0310 49 4E 50 50    03C0 41 01 E1 22
0088 A9 29 8D DD    0158 FB 85 F9 A0    0314 50 50 52 52    03C4 01 42 A0 A2
008C 01 A9 2C 8D    015C 01 B1 FA 10    0318 54 54 54 54    03C8 A1 C1 02 21
0090 DE 01 A9 59    0160 02 C6 F9 18    031C 54 54 53 53    03CC 61 4C 84 86
0094 D0 0C A9 2C    0164 69 02 90 02    0320 52 43 43 45    03D0 66 E6 C6 E0
0098 8D DD 01 A9    0168 E6 F9 18 65    0324 4D 4E 50 56    03D4 C0 24 20 78
009C 58 8D DE 01    016C FA 90 02 E6    0328 56 4C 4C 4C    03D8 03 03 03 02
00A0 A9 29 8D DF    0170 F9 20 3D 01    032C 4C 45 45 4E    03DC 01 02 02 02
00A4 01 A9 28 8D    0174 8D DC 01 8C    0330 4E 4F 48 48    03E0 02 02 03 02
00A8 DA 01 D0 17    0178 DB 01 A5 F9    0334 4C 4C 54 54    03E4 03 03 01 02
00AC A9 58 D0 02    017C 20 3D 01 8C    0338 41 41 53 58    03E8 C6 B9 B0 80
                                                        03EC 84 DA 7B 88
                                                        03F0 96 C6 C6 AC
                                                        03F4 AC B0 C6 7B
                                                        03F8 E9 ED ED ED
```

Bild 3. 6502-Maschinencode des Programms. Er enthält auch die erforderlichen Tabellen zur Decodierung der Programmbefehle

Schach dem Computer

Mikrocomputer-Schachspiel

Das Schachspiel, oft „Spiel der Könige" genannt, ist seit jeher eine der größten Herausforderungen an Computer-Hersteller und -Programmierer. Im Gegensatz zu nahezu allen anderen Spielen ist es beim Schach praktisch nicht möglich, alle Züge und damit die Möglichkeiten des Spielgegners vorauszuberechnen. In erster Linie ist das natürlich ein Zeitproblem; einfache Computer- oder Mikrocomputer-Schachprogramme verzichten daher auch darauf, eine möglichst große Anzahl von Zügen im voraus zu kalkulieren. Statt dessen wird für alle momentan möglichen Züge der eigenen Figuren je eine Bewertungszahl ausgerechnet, die sich nach arithmetischen Regeln aus Angriffsmöglichkeit, Sicherheit der eigenen Figuren und Bewegungsspielraum zusammensetzt. Derjenige Zug, der bei dieser „Vorausschätzung" die höchste Bewertung erhält, wird ausgeführt.

Seit einiger Zeit ist nun ein mit dem Mikroprozessor 8080 bestücktes Mini-Schachspiel auf dem Markt, der „Chess Challenger". Es wird von der amerikanischen Firma Fidelity Electronics hergestellt und arbeitet, im Gegensatz zu vielen anderen Spielen, nicht zusammen mit einem Fernsehgerät zur Darstellung des Spielbrettes, sondern ist ein eigenständiges Gerät (*Bild*).

Ein Rechner-Tastenfeld erlaubt zusammen mit einer vierstelligen Siebensegment-LED-Anzeige die Berechnung des Computer-Gegenzuges. Zu diesem Zweck ist die Anzeige in zwei Stellenpaare unterteilt, die jeweils das „Von-Feld" und das „Nach-Feld" (FROM und TO) angeben. Die Figuren müssen dann von Hand auf dem Schachbrett verschoben werden.

Die Anzeige erfolgt nach dem beim Schach üblichen System, nämlich als Matrixfeld mit einem Buchstaben (A...H) und einer Ziffer (1...8). Die Buchstaben sehen in dem Siebensegment-Display zwar etwas seltsam aus (insbesondere das „G" ist leicht mit der Ziffer 9 zu verwechseln, aber letztere existiert hier ja ohnehin nicht); immerhin muß man aber keine Tabellen bemühen, um herauszufinden, welches Schachbrettfeld nun gemeint ist.

Die Schwierigkeiten mancher Mikrocomputer-Schachspiele (z. B. Microchess für den KIM) wurden beim Chess Challenger bereits überwunden; er beherrscht auch solche „Spezialzüge" wie die Rochade. Die *Tabelle* zeigt ein Spielbeispiel.

Der Challenger läßt sich überdies auf drei verschiedene Schwierigkeitsgrade einstellen, die sich vor allem auch in der Geschwindigkeit der vom Computer ausgeführten Züge unterscheiden. So braucht der Prozessor beim leichtesten „Level" etwa drei Sekunden, die höheren Schwierigkeitsgrade beanspruchen 10 s und 30 s.

Eine neue, verbesserte Version, die ab sofort lieferbar sein soll, verwendet die CPU Z 80 von Zilog, bietet 10 Spielstufen und verfügt auch über eine Briefschach-Stufe, in der für jeden Zug eine stundenlange Problemanalyse durchgeführt wird – Zeit ist ja beim Briefschach kein Problem, und daher können auch 3 024 000 Positionen überprüft werden! Diese neue Version wird allerdings etwas teurer als die bisherige 8080-Ausführung werden. Fe.

Tabelle: Schachspiel mit dem „Chess Challenger"

Der Computer spielt schwarz.

Weiß	Schwarz
e2–e4	e7–e5
f1–c4	g8–f6
g1–f3	b8–c6
f3–g5	d7–d5
c4–b3	f6–e4
g5–e4	d5–e4
0–0	f8–c5
d1–h5	g7–g6
h5–e2	d8–d4
g1–h1	c6–e7
f2–f4	0–0
f4–e5	c8–f5
e5–e6	f7–e6
c2–c3	d4–d6
b3–c2	d6–d5
d2–d4	c5–b6
c1–h6	f8–d8
b1–d2	e4–e3
c2–b3	d5–a5
e2–e3	e7–d5
e3–e5	a5–b5
e5–g7	matt

(0–0: Rochade)

Mikroprozessor als Schachgegner: Der „Chess Challenger"

Mikro-computer

Mit diesem Heft beginnt eine neue, ständige Einrichtung in der FUNKSCHAU, nämlich die Rubrik „Mikrocomputer". In jedem Heft finden Sie einige Seiten mit Informationen darüber, welche Systeme es für den Hobbyelektroniker heute schon gibt und was in Kürze auf den Markt kommt; Sie finden Programme für nützliche und spielerische Anwendungen gängiger Mikrocomputer-Systeme, erfahren auch, wie sich typische Mikroprozessoren voneinander unterscheiden, und können – falls Sie sich bisher noch nicht mit dieser Materie beschäftigten – an einem humorvoll illustrierten Einführungskurs in diese heute bereits nicht mehr wegzudenkende Technik teilnehmen.

Hans Werner

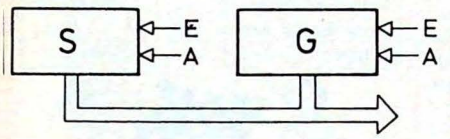
Mikrocomputer

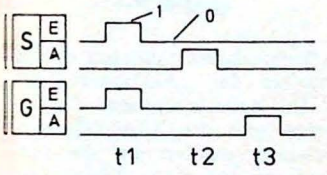
Eine kleine Einführung

2. Teil

Lösung der Aufgabe von Heft 21

Die Ausgabebefehle dürfen auf keinen Fall an beiden Speichern gleichzeitig liegen. So verbindet z. B. die Leitung D1 eine 0 mit einer 1. Und das ist ein Kurzschluß.





Zeitpunkt t1: Die Eingabe erfolgt gleichzeitig für beide Speicher. Sie darf natürlich auch nacheinander stattfinden.

Zeitpunkt t2: Ausgabe von Speicherinhalt S. Während des Impulses erscheinen die Daten auf den gemeinsamen Ausgangsleitungen.

Zeitpunkt t3: Ausgabe von Speicherinhalt G.

Solche gemeinsamen Datenleitungen werden „Datenbus" genannt. Die angeschlossenen Geräte sind mit Tri-State-Ausgängen ausgerüstet, die im hochohmigen Zustand keine Rückwirkung auf den Bus ausüben.

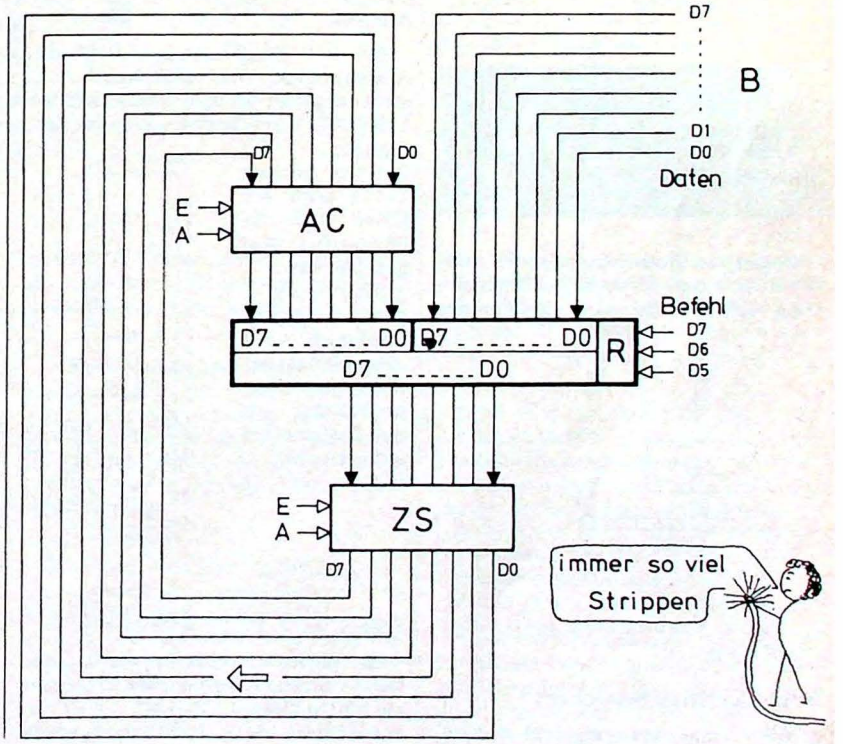
Der Akkumulator

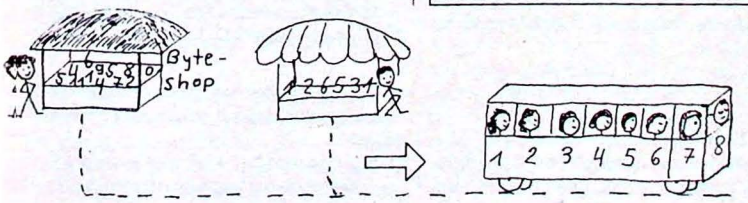
Mit den zwei erklärten Bausteinen kann der erste Teil des Modellcomputers entworfen werden.

Grundsätzlich sind alle Mikrocomputer Einadreßmaschinen, d.h. nur ein Eingang des Rechenwerkes ist nach außen geführt. Die genaue Bedeutung des Begriffes ist beim nächsten Entwurfsschritt besser zu erkennen.

Zu einer Rechnung gehören 2 Operanden A und B, die nacheinander über den einen Dateneingang an das Rechenwerk gelangen müssen. Dazu wird der 1. Operand durch das Rechenwerk geschoben und in einem speziellen Register, dem Akkumulator AC, abgelegt.

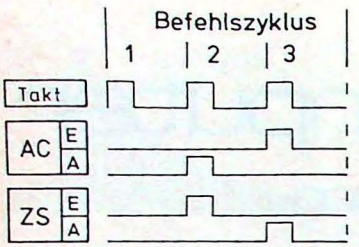
Der Akkumulator (Sammler) ist das Hauptregister eines Computers.





Beispiel:

Im AC steht der Wert A, am Dateneingang der Wert B. Als Befehl liegt 000 (AD) vor. Akkumulator und Zwischenspeicher werden durch einen Befehlszyklus von 3 Impulsen gesteuert.



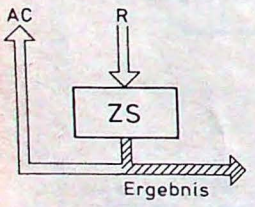
Impuls 1: Ist schon vorgesehen für den weiteren Aufbau und im Augenblick ohne Bedeutung.

Impuls 2: Der Akkumulator gibt den Wert A heraus, der im Rechenwerk zu B addiert wird. Gleichzeitig bekommt der Zwischenspeicher ZS seinen Eingabebefehl und übernimmt das Rechenergebnis C = A+B.

Impuls 3: Das Rechenergebnis wandert vom Zwischenspeicher in den Akkumulator (ZS, A und AC, E = 1). Dort steht es als Operand für den nächsten Rechenschritt zur Verfügung.

Im Akkumulator steht am Anfang der Rechnung der 1. Operand und nach der Ausführung des Befehles das Ergebnis.

Wegen des Zwischenspeichers spielen kurzzeitige falsche Rechenergebnisse keine Rolle. Das Endergebnis

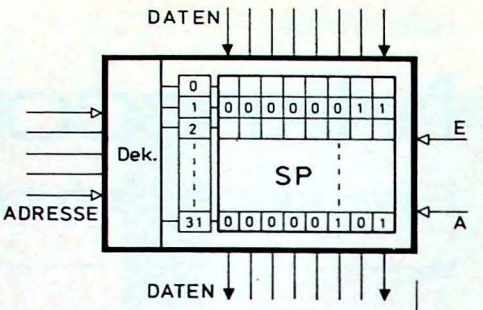


kann mit Impuls 3 nicht nur in den Akkumulator, sondern auch nach außen gegeben werden.

Adressierbarer Speicher

Sollen viele Daten abgelegt werden, so können entsprechend viele Speicher parallel auf einen gemeinsamen Ein- bzw. Ausgabebus arbeiten. Allerdings wird der Steueraufwand groß, denn jeder Speicher benötigt einen Eingabe- und Ausgabebefehl E und A.

Einfacher ist es, mehrere Speicher zusammenzufassen und die Auswahl



Adressierbarer Speicher

durch sogenannte Adreßleitungen vorzunehmen.

Unser Modellspeicher besteht aus 32 Speichern mit den Namen 0...31. Jeder Speicher hat 8 Zellen.

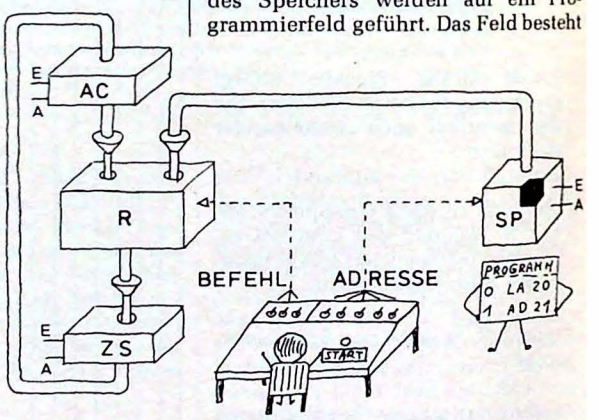
Zu den schon bekannten Daten- und Steuerleitungen kommen noch 5 Adreßeingänge, mit denen sich die Binärzahlen 00000 (0) bis 11111 (31) darstellen lassen. Ein Decodierer setzt die Binärzahlen in die Dezimalzahlen 0...31 um und spricht den gewünschten Speicherabschnitt an.

Beispiel:

An den Adreßleitungen liegt die Adresse 00001. Wird zusätzlich A zu 1, steht der Inhalt des Speicherabschnittes 1 (00000011) am Datenausgang zur Verfügung.

Mit der Adresse 11111 wird der Inhalt von 31 (00000101) ausgegeben usw.

Sind E und A gleich 0, werden wiederum alle Dateneingänge und -ausgänge hochohmig von den Speicherzellen getrennt.



Schnell noch einige Definitionen!

Ein **Bit** ist eine einzelne Stelle eines binär codierten Wortes, also die kleinste binäre Einheit.

Die Worte 101 und 100 haben jeweils 3 bit, unterscheiden sich aber nur in einem Bit.

Ein **Byte** besteht aus 8 bit. Der Modellspeicher hat somit eine Speicherkapazität von 32 Byte. Das sind 256 bit.

Ein KByte (Kilo-Byte) ist $2^{10} = 1024$ Byte und die Maßeinheit für große Speicher.

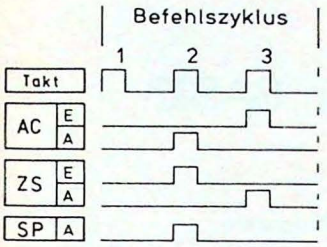
Ein adressierbarer Speicher erhöht den Komfort der „Akkumulatormaschine". Die Befehlsleitungen des Rechenwerkes und die Adreßleitungen des Speichers werden auf ein Programmierfeld geführt. Das Feld besteht aus 8 Schaltern. Jeder Schalter hat die Stellungen 0 und 1.

Befehlsfeld: 3 Schalter
Stellungen 000...111
Adreßfeld: 5 Schalter
Stellungen 00000...11111

Die Ausgabe des Rechenergebnisses kann hinter dem Zwischenspeicher erfolgen.

Ein Startknopf löst mit jedem Druck den Befehlszyklus aus (Impulse 1...3).

Die zugehörigen E- und A-Impulse steuern alle 4 Bausteine. Impuls 1 hat noch immer keine Bedeutung.



Programmbeispiel: Addition

Als Speicherzelle wird ab jetzt die kleinste adressierbare Einheit (1 Byte) bezeichnet.

In Speicherzelle
20 soll die Zahl 5 stehen: 00000101
In Speicherzelle
21 soll die Zahl 6 stehen: 00000110

Es sind 2 Befehlszyklen erforderlich. Es ist üblich, mit der Zahl 0 zu beginnen.

Befehlszyklus 0

Damit die Zahl 5 an den Eingang A des Rechenwerkes gelangt, muß der Inhalt der Speicherzelle 20 in den Akku geladen werden. Das Männchen stellt also 100 10100 (LA 20) am Schalterfeld ein und drückt auf den Startknopf.

Impuls 2: Der Speicher bekommt seinen Ausgabebefehl A, und der Inhalt von Zelle 20 steht am Rechenwerk. Durch den Ladebefehl LA (100) wird der B-Eingang des Rechenwerkes direkt zum Ausgang C geschaltet. Deshalb hat auch der Impuls AC, A keinen Einfluß. Gleichzeitig wird mit ZS, E die Zahl 5 (Inhalt von 20) in den Zwischenspeicher transportiert.

Impuls 3: Aus dem Zwischenspeicher gelangt die 5 in den Akku (ZS, A und AC, E = 1).

Befehlszyklus 1

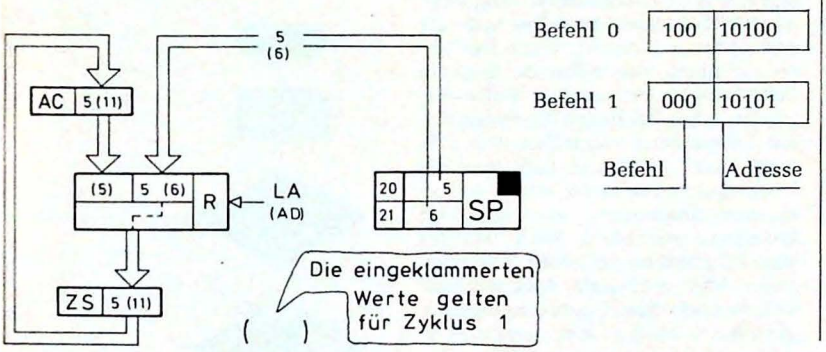
Der nächste Schritt ist die Addition 5+6. Die 6 steht in Zelle 21. Das Männchen stellt 000 10101 (AD 21) ein und betätigt wieder den Startknopf.

Impuls 2: Akkuinhalt (5) und Inhalt der Zelle 21 (6) gelangen zum Rechenwerk, werden addiert und im Zwischenspeicher abgelegt.

Impuls 3: Transport des Ergebnisses 5+6 = 11 in den Akku.

Zusammenfassung

Jeder Befehl besteht aus der eigentlichen Befehlsangabe und der Adresse. Es ist nur eine Adresse erforderlich (Einadreßmaschine).



Kürzel	Erklärung
0 LA 20	Lade den Inhalt der Zelle 20 in den Akku
1 AD 21	Addiere Inhalt Akku zu Inhalt Zelle 21

(Fortsetzung folgt)

µC-Nachrichten

Hitachi Ltd., will einen echten Heimcomputer auf den Markt bringen, der schon im Mai auf der „Good Living Show" in Tokio zu bestaunen war. Das Produkt ist etwa so groß wie eine Waschmaschine und besitzt einen eingebauten Mikroprozessor nebst Speichern und eine Tastatur. Ergonomisch ist der Heimcomputer auf die Bedürfnisse einer Hausfrau zugeschnitten, die damit ohne besondere Vorkenntnisse ihre gesamte Tagesarbeit organisieren kann. Zum Beispiel steuert der Computer das gesamte Heizungs-, Lüftungs- und Kühlsystem des Hauses, kontrolliert die Haushalts-Buchführung und erstellt bis zu 50 Tages-Speisepläne ohne und 200 mit einem zusätzlichen Magnetband. Zusätzlich sind geplant: Fensterverriegelung, Einbruchsicherung, Feuerschutz und Tierfütterung.

Wie die Firma Basis Mikrocomputer in Münster mitteilt, sind die Preise für den **Apple-II** und die zugehörige Peripherie um rund 25 % gefallen. Auch der Apple-II ist jetzt in Deutschland lieferbar. Als Neuheit wurde eine Spracheingabe vorgestellt, mit der er bis zu 32 gesprochene Worte verstehen kann.

Wayne Green, Herausgeber der US-Zeitschrift Kilobaud, schätzte die Mitte 1978 pro Monat verkauften Hobby- und Heimcomputer in den USA auf knapp 20 000; die Hälfte davon schrieb er dem TRS-80 zu, etwa ein Viertel dem PET 2001.

Als Weiterentwicklung der 6500-Mikrocomputer-Serie bringt Rockwell den **Einchip-Mikrocomputer 6500/1** heraus. Er wird in einem 40poligen Gehäuse geliefert und besitzt den gleichen Befehlssatz sowie alle Adressierungsarten wie die CPU 6502. Auf dem Chip befinden sich 2 KByte ROM, 64 bit RAM und ein 16-bit-Timer. Die 32 bidirektionalen E/A-Leitungen sind in vier 8-bit-Ports unterteilt, von denen einer interruptfähig ist. Ferner ist ein geeigneter Emulationsbaustein lieferbar, der 64 Pins besitzt und unter 200 DM kosten soll; der Preis für den 6500/1 liegt bei mittleren Produktionsmengen um 25 DM. Dieser Einchip-Mikrocomputer dürfte derzeit der schnellste erhältliche sein; derzeit verträgt er 1...2 MHz Taktfrequenz, ab Anfang 1979 sollen auch Versionen mit 3...4 MHz lieferbar sein.

Der Kieler Lichtkinetiker und Physiker Hans-Martin Ihme hat sein erstes **lichtkinetisches Kunstwerk** mit einem 6503-Mikroprozessor verwirklicht. Ein auf der Spitze stehender Würfel aus 720 Glühlampen sorgt für verblüffende Lichteffekte, die in einem 2-kByte-EPROM vorprogrammiert sind.

Synertek (6502-Second Source) und **Electronic 2000** (München) haben einen Distributor-Vertrag abgeschlossen, um Produkte wie ROM, RAM, Mikroprozessoren und Single-Board-Computer (VIM) in Deutschland zu vertreiben. Electronic 2000 hat nun auch in Düsseldorf ein Verkaufsbüro. Interessenten steht dort u. a. ein Mikrocomputersystem von Intel nebst System-Spezialisten zur Verfügung. Die Verkaufsleitung Nord übernimmt Ing. (grad.) Heinz G. Pape.

Rudolf Hofer

Obwohl es sich beim Biorhythmus um alles andere als eine auch nur halbwegs glaubwürdige Angelegenheit handelt, hat diese Pseudo-Wissenschaft in den letzten Jahren einen unglaublichen Aufschwung erlebt. Daß sie schon über 80 Jahre alt ist, ist den wenigsten bekannt.

Biorhythmus- und Wochentagsberechnung in BASIC

„Entdeckt" wurde der Biorhythmus von Dr. Fließ, einem Berliner Hals- und Nasenchirurgen, der ihn 1897 erstmals der Öffentlichkeit bekanntmachte. Dr. Fließ – der übrigens ein Jahrzehnt eng mit Sigmund Freud befreundet war – war davon überzeugt, daß das gesamte Leben zwei fundamentalen Zyklen (23 und 28 Tage) unterliegt. Später fand man noch einen weiteren von 33 Tagen. Den drei Zyklen werden die körperliche, seelische und geistige Verfassung zugeordnet. Sie beginnen mit der Geburt und laufen mit absoluter Regelmäßigkeit im Sinusformat ab (*Bild 1*). Die „kritischen Tage" ergeben sich laut Theorie, wenn der 23- oder 28-Tage-Zyklus die Horizontallinie schneidet. Ansonsten entsprechen die Amplituden dem jeweiligen Zustand.

Der Gregorianische Kalender

Das beschriebene Programm gibt nach Eingabe des Geburtstages und des Datums, für das die Prognose gewünscht wird, die Amplitude der drei Kurven zu diesem Zeitpunkt an. Wie leicht einzusehen ist, reduziert sich das Problem auf die Berechnung der Anzahl von Tagen zwischen Geburtstag und Prognosedatum mit anschließender Berechnung der Sinuswerte.

Um das zu erreichen, muß man zunächst etwas in der Geschichte blättern: Der heute gültige Gregorianische Kalender löste 1582 den Julianischen Kalender ab, da sich darin gegenüber dem Sonnenjahr geringe Verschiebungen ergaben. Auf den 4. Oktober folgte damals gleich der 15. Allerdings wurde diese Änderung von den Protestanten teilweise erst im 18. Jahrhundert übernommen.

Im Gregorianischen Kalender wird alle vier Jahre ein Schaltjahr eingefügt, das 366 Tage, also einen Tag mehr als normalerweise, dauert. Wenn die Jahreszahl durch vier teilbar ist, liegt ein Schaltjahr vor. Ausnahmen bilden diejenigen Jahre, bei denen die ersten beiden Ziffern durch vier teilbar sind, z. B. 1600, 2000, 2400 usw. Läßt man die letzte Bedingung außer acht – wie in unserem Programm – dann sind also Berechnungen vom 1. März 1600 bis zum 28. Februar des Jahres 2000 möglich. Mit geringem Mehraufwand könnte auch diese Einschränkung aufgehoben werden – aber warum übertreiben?

Das Programm

Das Hauptprogramm (*Bild 2*) nimmt Geburtsdatum und Prognosedatum entgegen und springt jeweils danach zu einem Unterprogramm, das die Anzahl der Tage von einem imaginären Zeitpunkt Null aus berechnet. Es spielt keine Rolle, wo dieser Nullpunkt liegt, da nur die Differenz zwischen Geburtstag und Prognosetag in die Berechnung eingeht.

Im Ausgabeteil (dargestellt durch das letzte Kästchen) wird noch einmal ein Unterprogramm verwendet, das aber im Flußdiagramm nicht gesondert aufgeführt ist. Es berechnet den Sinuswert der Differenz B-A (Zahl der Tage zwi-

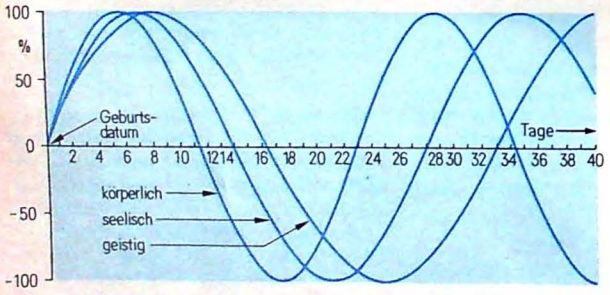


Bild 1. Verlauf der drei Zyklen

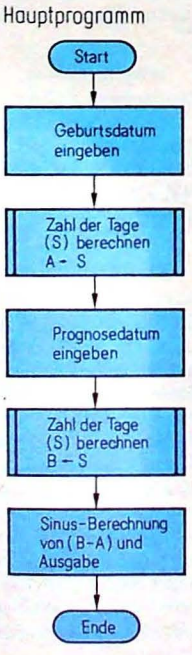


Bild 2. ▶ Flußdiagramm des Hauptprogramms

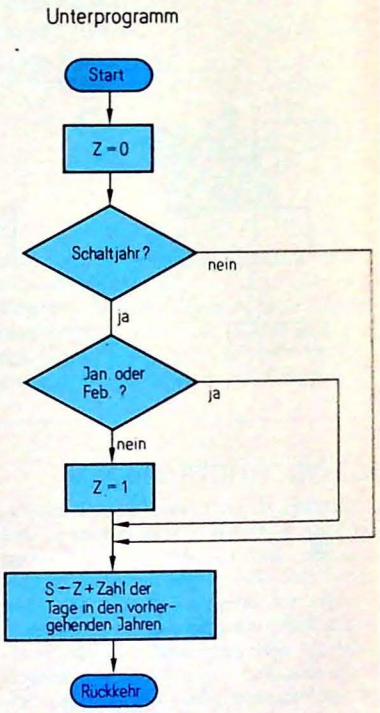


Bild 3. Flußdiagramm des Unterprogramms zur Berechnung der bisherigen Tage ab einem Bezugspunkt Null

```
0001 DIM M(12)
0010 PRINT "GEBURTSTAG":
0015 INPUT T,M,J
0020 GOSUB 100
0025  P=S
0027 RESTORE
0030 PRINT "PROGNOSE FUER":
0035 INPUT T,M,J
0040 GOSUB 100
0045  B=S
0050 PRINT "IHRE VERFASSUNG AM "STR$(T)"."STR$(M)"."STR$(J)" IST FOLGENDE: "
0052  L=28
0053  P=3.14159
 054 GOSUB 90
0055 PRINT "SEELISCH: "INT(N)"%"
0056  L=33
0057 GOSUB 90
0058 PRINT "GEISTIG : "INT(N)"%"
 059  L=23
0060 GOSUB 90
0061 PRINT "PHYSISCH: "INT(N)"%"
0070 GOTO 70
0090  N=100+SIN(2*P*(B-A)/L)
0095 RETURN
0100 REM UNTERPROGRAMM
0120  Z=0
0130 IF J/4<>INT(J/4)THEN300
0140 IF M<=2THEN300
0290  Z=1
0300  Z=Z+(J-1)*365+INT((J-1)/4
 0310 FOR K=1TO12
0320 READ M(K)
0330 NEXT K
 340  S=Z+M(M)+T
0350  R=S-INT(S/7)*7
0400 RETURN
0500 DATA 0,31,59,90,120,151,181,212,243,273,304,334

READY
```

◀ Bild 4. Biorhythmusprogramm mit Beispiel

```
0100 PRINT "TAG,MONAT,JAHR"
0110 INPUT T,M,J
0120  Z=0
0130 IF J/4<>INT(J/4)THEN300
0140 IF M<=2THEN300
0290  Z=1
0300  Z=Z+(J-1)*365+INT((J-1)/4
 0305 DIM M(12)
0310 FOR K=1TO12
0320 READ M(K)
 330 NEXT K
0340  S=Z+M(M)+T
0350  R=S-INT(S/7)*7
0370 FOR I=1TO7
0380 READ T$(I)
 390 NEXT I
0400 PRINT T$(R+1)
0500 DATA 0,31,59,90,120,151,181,212,243,273,304,334
0510 DATA SAMSTAG,SONNTAG,MONTAG,DIENSTAG,MITTWOCH,DONNERSTAG,FREITAG

READY
```

Bild 5. Wochentagprogramm mit Beispiel

schen Geburt und Prognosedatum) bei einer Periode von L Tagen, wobei der Variablen L nacheinander die Dauer der drei Zyklen zugeordnet wird. Mit 100 wird multipliziert, um Prozentangaben zu erhalten.

Das eigentliche Problem wird im ersten Unterprogramm (*Bild 3*) behandelt. Es liefert als Ergebnis den Wert S, der vom erwähnten Zeitpunkt Null aus die Gesamtzahl der Tage repräsentiert. Als Hilfsgröße wird Z eingeführt – die Tageszahl der vollen Jahre vor dem betrachteten Datum. Die Formel dafür (*Bild 4*) lautet Z = (J-1)·(365 + INT ((J-1)/4). J ist die Jahreszahl des betrachteten Datums, z. B. 1978. Da wir vorerst die Tage bis einschließlich 1977 wissen wollen, steht in der Formel J-1. Das wird mit 365 – der Tageszahl eines normalen Jahres – multipliziert. Nun fehlen noch die Schalttage, die genau ein Viertel von J-1 ausmachen, und zwar ohne Rest, also INT ((J-1)/4) (INT = Integer = Ganzzahl).

In Zeile 300 wird zu dieser Zahl noch ein vorher definierter Wert von Z (0 oder 1) addiert. Dieser Wert ist abhängig davon, ob das betrachtete Datum in einem Schaltjahr nach dem 29. Februar liegt. Dann nämlich muß die Tageszahl um eins erhöht werden. Die entsprechende Abfrage wird in den Zeilen 130 und 140 vorgenommen. Ist die Jahreszahl nicht ohne Rest durch 4 teilbar (also J/4 ≠ INT (J/4)), dann liegt kein Schaltjahr vor. Ist die Bedingung aber erfüllt, dann muß noch festgestellt werden, ob das Datum im Januar oder Februar liegt, also ob M ≦ 2 ist (M = Monat).

Um S zu erhalten, muß man jetzt noch die Anzahl der Tage im betrachteten Jahr (ohne 29. Februar, denn der ist schon berücksichtigt) ermitteln. Dazu werden den Monaten die bis zum Vormonat vergangenen Tage zugeordnet. Das geschieht in den Zeilen 310...330 und mit Hilfe der DATA-Zeile 500. Die indizierten Variablen M(1)...M(12) erhalten die Werte, 0, 31, 31 + 28, 31 + 28 + 31 usw. Der Wert M(7), der für Juli steht, ist also 181. Wer nachzählt, wird feststellen, daß bis zum Juli genau 181 Tage vergangen sind. Diese Zahl plus den jeweiligen Tag des Monats (T) ergibt demnach das gewünschte Ergebnis. In Zeile 340 wird diese Berechnung durchgeführt. Die Variable M(M) ist M(1) für Januar, M(2) für Februar usw. Die Variable M, die hier als Index verwendet wird, wurde gleich zu Anfang in der INPUT-Anweisung von Zeile 15 bzw. 35 eingegeben.

Drei Befehle aus dem Hauptprogramm sollten noch erwähnt werden: RESTORE in Zeile 27 sorgt dafür, daß auch beim zweiten Unterprogrammaufruf die DATA-Zeile wiederum von vorne gelesen wird. DIM M(12) in Zeile 1 erlaubt die Verwendung von zwölf Indizes, und GOTO 70 in Zeile 70 ist ein dynamischer Stop; er dient dazu, den Ausdruck READY am Programmende zu unterdrücken.

Berechnung des Wochentags

Mit leichten Änderungen kann das Programm auch den Wochentag eines beliebigen Datums innerhalb der eingangs erwähnten Grenzen bestimmen. Im wesentlichen wird dazu das beschriebene Unterprogramm verwendet. Es berechnet wiederum die Summe S, teilt diese durch sieben und bestimmt den Rest R (Zeile 350 in *Bild 5*) – man nennt das auch modulo 7. Auf diese Weise erhält man eine Zahl zwischen 0 und 6, die genau dem Wochentag entspricht. Nun ordnet man der String-Variablen den Index R zu und liest aus der DATA-Zeile der Reihe nach die Wochentage ein. Man muß nur einmal dafür sorgen, daß ein Datum richtig bestimmt wurde, dann stimmt auch jeder andere Tag. Daß in Zeile 400 PRINT T (R+1) steht, hat den Grund, daß im verwendeten BASIC der Index 0 verboten ist.

Beide Programme behandeln fehlerhafte Eingaben nicht gesondert. Wegen des hohen Aufwands im Verhältnis zur Programmlänge wurde darauf verzichtet.

Literatur

[1] *Gardner, M.:* Mathematischer Karneval. Ullstein-Verlag.

Der kalifornische Hersteller Hydro Products hat die Produktion eines „Unterwasserfahrzeugs" für Explorationszwecke aufgenommen. Es ist wahrscheinlich das zur Zeit „intelligenteste" Fahrzeug dieser Art, das kommerziell oder in der Forschung im Einsatz ist.

Mikrocomputer steuert Unterwasserfahrzeug

Die gesamte Datenübermittlung sowie alle Anzeige- und Kontrollfunktionen des Fahrzeugs werden durch zwei identische 16-bit-Prozessoren vom Typ Pace von National Semiconductor gesteuert. Einer der beiden Mikrocomputer befindet sich im Unterwasserfahrzeug, der zweite in einer bemannten Kontrollstation an der Oberfläche.

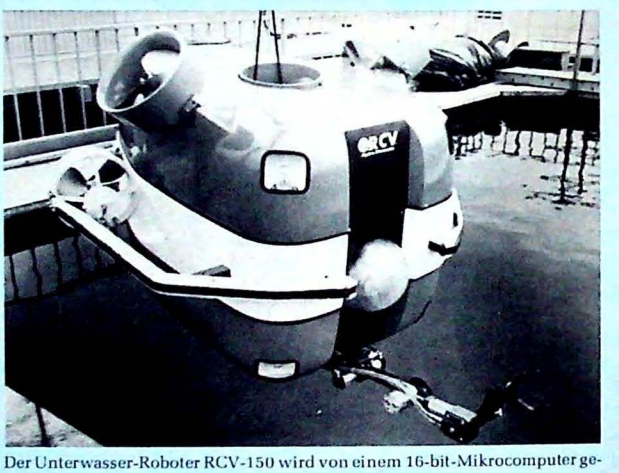
Ausgestattet mit Fernsehübertragungseinrichtung, Manipulierarm, Sonarsystemen und einer Reihe anderer Sensoren erfreut sich der Unterwasser-Roboter RCV-150 von Hydro Products (*Bild*) lebhafter Nachfrage. Sechs Aufträge für das 400 000-Dollar-Basissystem lagen vor, als sich das Projekt noch im Konstruktionsstadium befand. Die wichtigsten Anwendungen bestehen etwa darin, Standorte für Ölbohrungsplattformen ausfindig zu machen, Unterwasserpipelines zu inspizieren oder Objekte auf dem Meeresboden zu suchen.

Der Unterwasserteil des RCV-150 besteht aus einem per Kabel geführten Tauchkörper mit den Abmessungen 120 cm x 105 cm x 120 cm. Er wird von vier 25-cm-Schrauben angetrieben. Die Flexibilität des RCV-150 beruht auf dem Ein-/Ausgabesystem des Mikrocomputers, das den Anschluß von 16 Sensoren ermöglicht. Wird eine weitere 16-Kanal-Platine eingesteckt, läßt sich das System leicht erweitern. Da vorläufig nur die Hälfte des verfügbaren Platzes genutzt wird, sind Sondereinbauten wie Magnetometer, Sonartiefenmesser, 35-mm-Kameras oder ein zweiter Greifarm möglich.

Alle Programme zur Bewegung des Fahrzeugs in drei Achsen – in Abhängigkeit von Fahrtrichtung, Tiefe, Temperatur usw. – laufen unter Kontrolle des Pace-Mikrocomputers. Das befreit den Operator von Lenkaufgaben und gibt ihm die Möglichkeit, sich auf Beobachtung und Inspektion sowie auf die Manipulation des Greifarms zu konzentrieren. Da das Fahrzeug Tiefe und Fahrtrichtung automatisch kontrolliert, kann es in Strömungen seine Position beibehalten, ohne daß der Operator eingreifen muß.

Neun Monate dauerte die Entwicklung, weitere 1,5 „Mannjahre" wurden für die Programmentwicklung benötigt. Dabei zielte die Entwicklung darauf, möglichst viele Systemeigenschaften per Software zu realisieren.

Zur Standardausrüstung gehören Sensoren zur Erfassung der Fahrzeugbewegung, Temperaturfühler und Detektoren zum Erkennen von eingedrungenem Seewasser. Sehr vielseitig einsetzbar ist auch der 1 m lange Manipulierarm, der in vier Richtungen beweglich und mit einer Reihe von Werkzeugen ausgerüstet ist.



Der Unterwasser-Roboter RCV-150 wird von einem 16-bit-Mikrocomputer gesteuert

Die Leitstation enthält zwei Monitore zur Wiedergabe des Kamerabildes und zur Anzeige der wichtigsten Daten wie Tiefe, Fahrtrichtung, Kameraeinstellung und Temperatur. Analoge Instrumente geben Auskunft über die Antriebsleistung, die Temperatur im Elektronikgehäuse und über die Erwärmung des Hydrauliköls. Eine alphanumerische Tastatur erlaubt dem Operateur die Eingabe von Daten in das System. Mit zwei Steuerknüppeln ist die Steuerung des Fahrzeugs und der Kamera möglich.

Zehnmal pro Sekunde fragt der Mikrocomputer in der Leitstelle die Bedienungskonsole ab. Gibt der Operator ein Steuerkommando, wird das entsprechende Signal zur Multiplexübertragung codiert und über Hf-Telemetrie-Modem auf Koaxialkabel zu einem gleichartigen Modem im Fahrzeug übermittelt. Der Mikroprozessor im RCV 150 interpretiert das Kommando und aktiviert die entsprechenden elektrohydraulischen Servos, wobei er simultan laufend die erforderlichen Bewegungsgleichungen durchrechnet, und zwar 30mal je Sekunde.

Von den Sensoren und Detektoren des Fahrzeugs aufgenommene Daten werden in umgekehrter Richtung zum Leitstellencomputer übertragen, der sie an die zugehörigen Anzeigeeinrichtungen auf dem Bedienungspult weiterleitet. Die Übertragung von Telemetriedaten erfolgt durch ein Vierkanal-Multiplexsystem über ein einzelnes Koaxialkabel.

Ursprünglich hatte man daran gedacht, in der Leitstelle einen Minicomputer zur Lösung der Bewegungsgleichungen und Übertragung der zum Betrieb der Servos erforderlichen Daten einzusetzen. Genauere Überlegungen zeigten jedoch, daß sich mit dieser Methode die 30 erforderlichen Berechnungen pro Sekunde nicht durchführen ließen, vor allem, weil das Übertragungssystem diese hohe Datenübertragungsrate nicht verarbeiten konnte. Die Verwendung eines Mikrocomputers im Unterwasserfahrzeug löste dieses Problem.

Um langsam ablaufende Vorgänge, z.B. die Entladekurve eines Nickel-Cadmium-Akkus, darzustellen, ist die Verwendung eines Speichervorsatzes in Verbindung mit einem handelsüblichen Oszilloskop recht praktisch. Wie ein solcher Speichervorsatz mit dem Microcomputer KIM-1 realisiert werden kann, erfahren Sie im folgenden Beitrag.

Oszilloskop + KIM = Speicheroszilloskop

Das Abspeichern einer analogen Information ist leider nicht ohne weiteres möglich; ein Mikrocomputer kann schließlich nur digitale Bits oder Bytes speichern. Aus diesem Grunde ist zunächst einmal ein Analog-Digital-Wandler erforderlich, der die zu messende Eingangsspannung in ein digitales Wort, ein 8-bit-Byte, umsetzt.

Analog-Digital-Wandler sind nun leider recht teuer, und da man zur Wiedergabe der gespeicherten Spannungswerte ohnehin einen Digital-Analog-Wandler benötigt, wurde eine preisgünstigere Lösung verwendet.

Sie besteht darin, den Digital-Analog-Wandler durch zwei Operationsverstärker und etwas Software auch als Analog-Digital-Umsetzer anzuwenden. Dazu dient die in *Bild 1* gezeigte Schaltung. Der Mikrocomputer ändert gezielt die am D/A-Wandler anstehenden Daten (KIM-Port A) solange, bis dieser eine der zu messenden Eingangsspannung entsprechende Spannung liefert. Dies geschieht nach einem Algorithmus, der ähnlich wie das bekannte Zahlenratespiel Hi-Lo arbeitet und eine Umsetzzeit von nur 200...300 µs für ein 8-bit-Wort ermöglicht. *Bild 2* zeigt das gesamte Programm in hexadezimaler Form; der Teil im Adressenbereich 00C5...00E8 dient der A/D-Umsetzung. (Zur Orientierung enthält Bild 2 einige Adressenangaben; Startadresse ist 0000.)

Die Bedienung dieses Mikrocomputer-Speicherzusatzes ist denkbar einfach. Sobald das Programm an der Adresse 0000 durch Drücken der Taste GO gestartet wurde, zeigen alle sechs Siebensegment-Displays des KIM-1 eine Null. Mit den Tasten 0...9 läßt sich nun eine beliebige Aufzeichnungsdauer eingeben; hat man sich vertippt, kann man mit der Stop-Taste (ST) die Anzeige wieder löschen. Es lassen sich Zeiten zwischen 1 s und 999 999 s eingeben.

Die Aufzeichnung des analogen Spannungsverlaufes am nichtinvertierenden Eingang des Operationsverstärkers 748 beginnt, sobald nun die Taste AD („Analog/Digital") gedrückt wird. Für die gerade gewählte Aufzeichnungszeit bleibt das Display dunkel und zeigt dann wieder 000 000 an. Mit der Taste DA (Digital/Analog) lassen sich nun die 512 Bytes des Speicherbereiches 0200...03FF wieder abrufen und mit etwa 18 ms Wiederholzeit auf dem angeschlossenen Oszilloskop darstellen. Das Programm sorgt auch automatisch dafür, daß zu Beginn jedes Zyklus ein Triggerimpuls zur Synchronisation des Oszilloskops mit vollem Spannungshub (U_{ss} = 5,12 V) erzeugt wird.

Aus dem Darstellungs-Modus kann durch Drücken von ST wieder herausgesprungen werden. Diese Taste ermöglicht auch ein vorzeitiges Abbrechen einer Aufzeichnung. Während das Display leuchtet, kann schließlich mit der Taste C der gesamte Speicherinhalt (Adressenbereich 0200...03FF) gelöscht werden.

Die Auflösung dieses „Speicheroszilloskops" beträgt 256 Amplitudenschritte (vertikal) und 512 Zeitschritte (horizontal). Für die beiden Operationsverstärker empfiehlt sich eine Betriebsspannung von etwa ±12 V.

H. Feichtinger

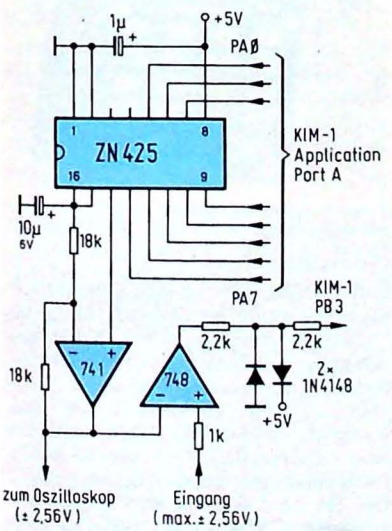


Bild 1. Notwendige Hardware für das „Speicheroszilloskop"

```
a2 ff       0061 ao oo      00c5 a9 ff
9a               84 fa           8d o1 17
a9 oo            a9 o2           8d oo 17
8d fa 17         85 fb           a9 8o
8d fb 17         a9 ff           aa
85 f9            8d o1 17        4d oo 17
85 fa            8d oo 17        8d oo 17
85 fb            a9 oo           a9 o8
85 f6            8d oo 17        2c o2 17
a9 o2            b1 fa           fo o7
85 f7            8d oo 17        8a
2o 1f 1f         2o 63 1f        4d oo 17
do fb            a5 fb           8d oo 17
2o 1f 1f         c9 o4           8a
fo fb            do f2           4a
2o 1f 1f         fo db           9o e7
2o 6a 1f         a9 oo      00e8 4c oo o1
c9 1o            85 f3
fo 5b            85 f4      o1oo ad oo 17
c9 11            85 f5           ao oo
fo 32            d8              91 f6
c9 oc            a5 f3           e6 f6
fo 16            c5 f9           do ob
c9 oa            do oc           e6 f7
1o eo            a5 f4           a5 f7
2a               c5 fa           c9 o4
2a               do o6           do o3
2a               a5 f5           4c oo oo
2a               c5 fb           4c 86 oo
ao o4            fo 24      o119 ende
2a               a9 1e
26 f9            8d o6 17
26 fa            2c o7 17
26 fb            1o fb
88               f8
do f6            18
fo ce            a5 f3
a9 o2            69 o1
85 fb            85 f3
a9 oo            9o d9
85 fa            a5 f4
ao oo            69 oo
91 fa            85 f4
2o 63 1f         9o f6
a6 fb            a5 f5
eo o4            69 oo
do f5            85 f5
4c oo oo         9o ee
```

Bild 2. Auflistung des Programms im 6502-Maschinencode

Magnetblasen-Speicher ab 1979 lieferbar

Texas Instruments kündigte kürzlich an, ab dem ersten Halbjahr 1979 einen 0,25-MByte-Magnetblasen-Speicher zusammen mit den dafür erforderlichen peripheren IS liefern zu können, der dann mit den Mikrocomputer-Familien 9900 und zum Teil auch 8080 kompatibel sein wird.

Der Blasenspeicher TB 0303 arbeitet mit Domänen von 3 µm und besitzt eine Nebenschleifen-Organisation mit Blockreplikation sowie getrennte Schreib- und Lesespuren. Insgesamt 252 Nebenschleifen mit 1137 Blasenpositionen ergeben eine Speicherkapazität von 286 524 bit auf einem einzigen Chip!

Die Datenbits werden in die Schreibspur eingeschrieben und über Austausch-Gatter mit den gespeicherten Daten in den Nebenschleifen ausgetauscht.

Bei 100 kHz Taktrate ergibt sich eine durchschnittliche Zugriffszeit von 7,3 ms für das erste Bit einer 224-bit-

Seite. Die Leistungsaufnahme beträgt rund 0,9 W. Wegen der Verschachtelung der Datenschleifen entspricht die Lese- und Schreibrate der Taktrate, nämlich 100 kByte pro Sekunde. Die Betriebstemperatur reicht von 0 °C bis 50 °C, und eine Lagerung ohne Verlust des Speicherinhalts ist von −40 °C bis 85 °C möglich.

Eine Erprobungsplatine mit einem solchen Bubble-Speicher und den erforderlichen Peripherie-Schaltkreisen wird ebenfalls im ersten Halbjahr 1979 verfügbar sein. ■

PET für Industrieanwendungen

Einen interessanten Anwendungsbereich für den PET 2001 (s. FUNKSCHAU 12/1978) hat die Fa. *NIDOS* (Nuclear and Industrial Data Organisation System) erschlossen: die Prozeßsteuerung. NIDOS ist ein modulares System, das es ermöglicht, Prozesse sowohl vollautomatisch, z. B. über einen externen Rechner, halbautomatisch oder manuell zu steuern. Dank des modularen Aufbaus des Systems, das vor allem für die Prozeßkontrolle und Datenerfassung in der Industrie, der Kernforschung und der Medizin entwickelt wurde, ist es möglich, schnell und flexibel auch komplexere Prozesse zu automatisieren. Das System besteht bereits aus einer Reihe von Standard-Moduln wie Timern, Zählern, Analog-Digital-Konvertern, PID-Reglern, Digital-Analog-Konvertern, Relaismoduln, Schrittmotorsteuerungen und Meßstellenumschaltern, die sich in einem 19"-Überrahmen mit der Stromversorgung und dem NIDOS-Datenbus befinden. Durch den modularen Aufbau ist es möglich, kundenspezifische Sonderentwicklungen in das bestehende Standardprogramm zu integrieren. Die weitgehende Verwendung von CMOS- und Low-Power-Schottky-Schaltkreisen, Alu-Gehäusen und Industriesteckern ergeben einen guten Störabstand (die Module können ohne weiteres während des Betriebes ausgewechselt werden), eine hohe Betriebssicherheit und geringe Wärmeentwicklung.

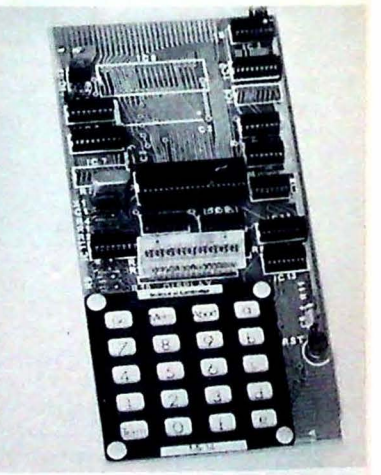
Die Besonderheit des NIDOS-Systems ist, daß man jede Funktion auch bei vollautomatischem Betrieb jederzeit am Modul auf „Hand" umschalten kann und damit unverzüglich in den Prozeßablauf eingreifen kann. Diese Konzeption ermöglicht es, den Prozeß auch bei Ausfall des Rechners unter Kontrolle zu halten oder schnell Parameter zu ändern.

Eine bittere Pille zum Schluß: Der Preis des Systems beträgt natürlich ein Vielfaches von dem, was man für einen PET hinblättern muß. ■

Preiswertes SCMP-Mikrocomputersystem

Seit kurzer Zeit ist ein Mikrocomputer im Handel, der weniger als 200 DM kostet und doch alles bietet, was man von einem einfachen System erwarten kann: Tastatur für sedezimale Eingabe von Adressen und Daten, ein Siebensegment-LED-Display und ein im ROM stehendes Betriebsprogramm namens „KITBUG".

Das mitgelieferte Handbuch enthält nicht nur die nötigen Anweisungen für den Zusammenbau des Kits (es wird mit einer fertig geätzten und gebohrten Platine geliefert), sondern auch eine Auflistung des Betriebsprogramms in sedezimaler Form und in Assemblersprache, ferner einige Programmbeispiele mit Multiplizieren, zum Dividieren, zur Wurzelberechnung und zum Errechnen des größten gemeinsamen Teilers einer Zahl als Mathematik-Paket. Weitere Programme machen aus dem Mikrocomputer eine Digitaluhr mit Wecker, erzeugen weißes Rauschen an einem Ausgang, ermöglichen den Verkehr mit einem Fernschreiber, oder wandeln Zahlen verschiedener Systeme um. Im Handbuch enthalten sind auch Spiele wie Mondlandung, Entenjagd, Mastermind, Silberdollars. Mit einem 8-bit-D/A-Wandler (rd. 10 DM) läßt sich der Mikrocomputer schließlich noch als Funktionsgenerator mit beliebigen Ausgangs-Wellenformen und als Einfachst-Elektronenorgel einsetzen.



Mikrocomputer für weniger als 200 DM: Der MK-14 mit dem Mikroprozessor SCMP (Eltec-Elektronik, Mainz)

Der zur Verfügung stehenden Speicherplatz ist 256 Byte (RAM), läßt sich aber selbstverständlich erweitern, wofür bereits Platz auf der Platine vorhanden ist. 16 I/O-Ports sind bereits herausgeführt, ebenso ein serieller Ausgang und die Flag- und Sense-Anschlüsse des SCMP-Mikroprozessors.

16-bit-CPU von Intel

Von Intel ist jetzt eine 16-bit-Mikrocomputer-Familie zu haben. Kernstück ist die CPU 8086; sie hat praktisch alle Eigenschaften des bekannten 8080, verfügt zusätzlich aber über eine 8-bit- und 16-bit-Arithmetik einschließlich Multiplikation und Division. Die CPU kann dank der größeren Wortlänge nun 1 MByte direkt adressieren. Die Architektur des Chips erlaubt auch Multiprozessor-Betrieb.

Der Datendurchlauf ist wegen der höheren Taktfrequenz und der 16-bit-Verarbeitung rund 7mal schneller als bei der CPU 8080 A. Die Programme können außerdem dank des etwas verbesserten Befehlssatzes um 10...20 % verkürzt werden.

Die Bus-Interface-Einheit beinhaltet u. a. ein 6 Byte langes FIFO-Register (First In, First Out) als Buffer für das nächste Befehlswort. Damit ist eine Anpassung der Prozessorgeschwindigkeit an die Busgeschwindigkeit möglich, ohne bei direktem Speicherzugriff an Geschwindigkeit zu verlieren.

An Sprungbefehlen und Unterprogramm-Aufruf-Befehlen gibt es zwei Arten, je nachdem, in welchem Adressenbereich gearbeitet wird. Die eine Gruppe läßt Sprünge innerhalb eines beliebigen 64-k-Segments zu, während die andere Gruppe einen indirekten Sprung über einen 32 bit langen Zeiger ausführt, der irgendwo im Speicher liegen kann. Sprünge innerhalb eines Segments sind stets relativ, was der „Relocation", der Übertragung eines bereits geschriebenen Programms in einen anderen Speicherbereich, zugute kommt. ■

Tests sollten nicht übersprungen, aber auch nicht tierisch ernst genommen werden. Sie sind die „Wachmacher" im Lehrstoff. Die Intelligenz kann und soll nicht getestet werden.

Hans Werner

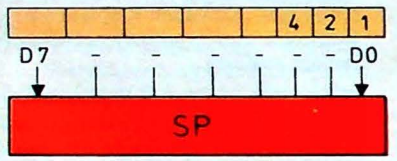
Mikrocomputer

Eine kleine Einführung

3. Teil

a) „nach K" bedeutet: Die Lösung ist unter Rubrik K zu finden.

b) „nach N" bedeutet: Wenn Sie Lösung b wählen, schlagen Sie unter N nach.



Jedem Eingang D0...D7 ist eine binäre Wertigkeit zugeordnet. Füllen Sie die freien Felder aus.
Die größte darstellbare Zahl ist...

nach K

A

Das ist leider nicht richtig. In der Digitaltechnik treten grundsätzlich nur zwei Spannungspegel auf (z.B. 0 V und 5 V). Tri-State-Ausgänge haben zwar drei Zustände, aber einer davon ist der hochohmige Zustand (Ausgänge abgeschaltet).

nach H

B

Ein KByte hat $2^{10} = 2 \cdot 2 \cdot 2 \cdot 2 \cdot 2 \cdot 2 \cdot 2 \cdot 2 \cdot 2 \cdot 2$ = 1024 Byte. Das sind 8 · 1024 = 8192 bit. Es waren also alle Antworten richtig.

nach D

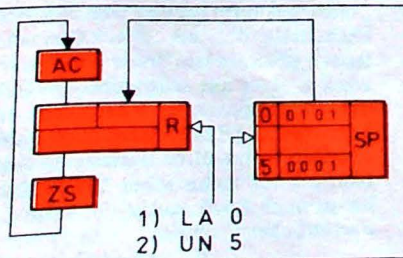
C

Ein Speicher hat 3 Adreßeingänge. Wie groß ist seine Speicherkapazität, wenn mit jeder möglichen Adreßkombination ein Byte angesprochen wird?

☐ a) weiß ich nicht
☐ b) 16 Byte
☐ c) 64 Bit

bei Lösung
a) nach I
b) nach N
c) nach S

D



Es werden nacheinander die Befehle 1) und 2) ausgeführt.

Was steht nach jedem Befehl im Akku?

☐ a) nach LA 0: 0101
nach UN 5: 0001

☐ b) nach LA 0: 0000
nach UN 5: 0001

bei Lösung

a) nach O
b) nach F

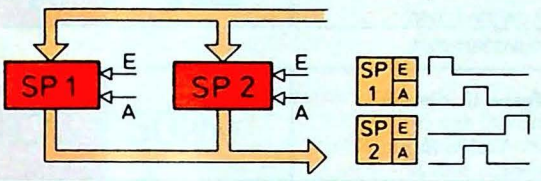
E

Sie haben etwas verwechselt. Mit dem Befehl LA 0 wird nicht die Zahl 0, sondern der Inhalt der Zelle 0 in den Akku transportiert.

nach P **F**

Sehr gut, die Lösung ist richtig.

nach H **G**



Dürfen beide Speicher mit dieser Impulsreihe gesteuert werden?

☐ a) Ja
☐ b) Nein

bei Lösung
a) nach M
b) nach Q

H

000	100
001	101
010	110
011	111

3 Adreßeingänge lassen $2^3 = 8$ Kombinationen zu (siehe Tabelle). Damit hat der Speicher eine Kapazität von 8 Byte = 64 bit.

nach E **I**

Das war sicherlich ein Versehen. Versuchen Sie es nochmals.

nach L **J**

128 64 32 16 8 4 2 1
D7 D0

Die größte darstellbare Zahl ist $1 + 2 + 4 + \ldots 128 = 255$.

nach L **K**

Was sind Tri-State-Ausgänge?

☐ a) wenn z. B. die 3 Spannungspegel 0 V, 5 V, 10 V auftreten.
☐ b) wenn z. B. die 3 Zustände 0 V, 5 V, hochohmig auftreten.
☐ c) der Ausdruck wird für a) und b) benutzt.

bei Lösung
a) nach B
b) nach G
c) nach J

L

Falsch! – Die Datenausgabe erfolgt bei beiden Speichern gleichzeitig. Dann dürfen sie nicht an einen gemeinsamen Ausgabebus angeschlossen sein.

nach R **M**

Das war bestimmt ein Versehen. Lesen Sie bitte die richtige Lösung.

nach I **N**

Ausgezeichnet! Die Antwort ist richtig.

nach P **O**

Jeder Befehl des Modellcomputers besteht aus 8 bit und teilt sich in Operationsteil (AD, SU...) und Adreßteil auf. Der Operationsteil hat 3 bit. Damit mehr als $2^3 = 8$ Befehle möglich sind, wird er um 1 bit erweitert. Die Gesamtzahl von 1 Byte bleibt erhalten. Was hat das für einen Nachteil?

nach T **P**

Richtig! Sie haben erkannt, daß die Ausgabe nicht gleichzeitig erfolgen darf.

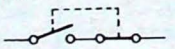
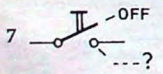
nach R **Q**

Welche Antwort ist richtig? Ein KByte hat ☐ a) 1024 Byte ☐ b) 2^{10} Byte ☐ c) 8192 bit	nach C	R
Sie haben recht. Es sind 8 Byte = 64 bit.	nach E	S
Für den Adreßteil bleiben nur noch 4 bit übrig, d. h. es können nur noch 2^4 = 16 Speicherplätze adressiert werden. Das war die letzte Aufgabe in diesem Test. Vielen Dank!	PAUSE	T

Tante Olgas Rätselecke

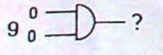
■	1	2	3			4	5
6							
7					8		
				9			
	■	■	10				■

Senkrecht:

1 logische Verknüpfung
2 große Rechenanlage
3 ...immer Treu und Redlichkeit
4 verliert ein Speicher nach dem Auslesen seine Information?
5 Fachausdruck für die Schaltung 
7 

8 Nein

Waagerecht:

3 1 v 1 = 1
6 von -Zyklus
7 log. Bedeutung von +
8 soll die Potenz stärken
9 

10 elektr. geladenes Teilchen

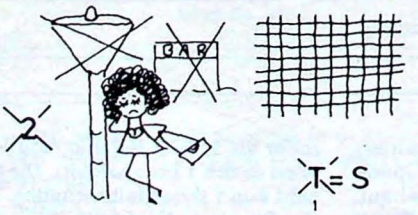
Keineswegs soll es eine Pflichtübung sein, Rätsel zu lösen.

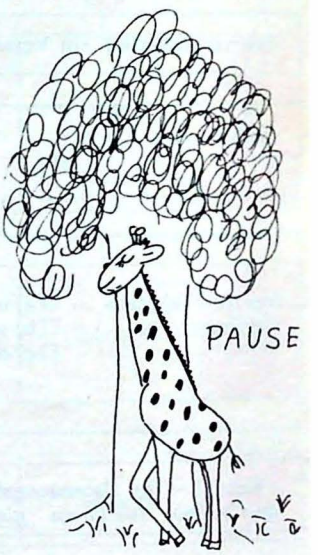
Regel: Wer will, der tut

Bemerkungen zu den Kreuzworträtseln: Die deutsche Sprache hat eine Redundanz von 66 %, d. h. sehr viele Buchstabenkombinationen ergeben kein sinnvolles Wort. Soll sich das Kreuzworträtsel noch zusätzlich auf einfache Ausdrücke der Digitaltechnik beschränken, wird es dementsprechend primitiv. Der Leser mag das entschuldigen.

Bilderrätsel

Zu erraten ist ein Fachbegriff mit 8 Buchstaben. Die Bilder symbolisieren 2 bzw. 6 Buchstaben des Wortes.





Zahlenrätsel

Die Aufgabe 1 + 2 = 11 erscheint eigenartig.

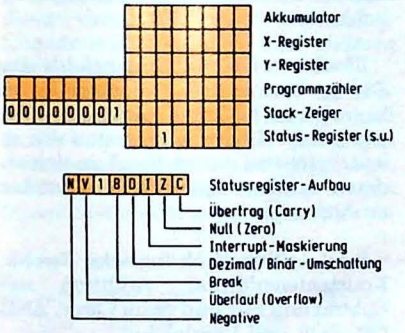
Wo liegt der Trick?

Fortsetzung folgt

Sie finden in der FUNKSCHAU recht oft Programme für den Mikroprozessor 6502. Für diejenigen Leser, die sich mit diesem µP bisher noch nicht beschäftigt haben, bringen wir eine kurze Übersicht der wesentlichsten Eigenschaften dieses auf dem Hobbycomputer-Markt inzwischen recht verbreiteten Prozessors.

MCS 6502

Der MCS 6502 wurde von MOS Technology entwickelt; diese Firma ist mittlerweile von Commodore (PET 2001) übernommen worden. Als Zweitlieferanten fungieren derzeit Synertek und Rockwell. Der Prozessor ist das Mitglied einer ganzen Familie von µPs,



Aufbau der internen CPU-Register beim MOS 6502

zu der auch die Typen 6503, 6504, 6505 und 6506 gehören, die aber bisher weniger Verbreitung fanden.

Die 65XX-Familie ging ursprünglich durch Weiterentwicklung aus der Motorola-6800-Familie hervor. Die Entwicklungsphilosophie war dabei eine gänzlich andere als z. B. bei den Prozessoren 8080 oder Z 80, und ein Vergleich mit diesen beiden ist daher nur schwer möglich.

Das beginnt bereits bei dem internen „Timing", dem Zeitablauf. Alle Befehle werden in ganzzahligen Vielfachen der Takt-Periodendauer ausgeführt, und viele Befehle brauchen nur ebenso viele Taktzyklen, wie sie Bytes benötigen! Die CPU 6502 ist dadurch einer der schnellsten Prozessoren, die derzeit erhältlich sind, obwohl die übliche Taktfrequenz „nur" 1 MHz und maximal 2 MHz beträgt.

Das *Bild* zeigt die im Prozessor vorhandenen Register und ihre Wortlänge; addieren und subtrahieren kann man nur im Akku, und zwar – je nachdem, ob das Dezimal-Bit im Statusregister 1 oder 0 ist – im Dezimal- oder im Binärsystem. Dadurch ist der bei dem Prozessor 8080 vorhandene Befehl „Decimal Adjust" hier nicht erforderlich.

Außer für Zwischenspeicherungen können die X- und Y-Register für die besonderen Adressierungsarten des MCS 6502 verwendet werden. Diese sind:

Adressierung	*Beispiel*	*Wirkung im Beispiel*
Immediate	A 9 02	Akku wird mit dem Wert 02 geladen
Absolute	AD 00 17	Akku wird mit dem Inhalt der Zelle 1700 geladen
Zero Page	A 5 08	Akku wird mit dem Inhalt der Zelle 0008 geladen
Indirect	6C 67 02	Sprung zu der Adresse, die in den Zellen 0267 und 0268 steht
Implied	CA	X-Register um 1 erniedrigen (der Befehl enthält die Adressierung selbst)
Abs., indiziert	BD 00 02	Der Akku wird mit dem Inhalt der Adresse geladen, die sich aus der Summe der angegebenen (0200) Adresse und dem X-Register ergibt
Nach-indiziert	B1 FA	Der Akku wird mit dem Inhalt der Zelle geladen, die sich aus der in den Zellen 00FA und 00FB stehenden Adresse plus dem Inhalt des Y-Registers ergibt
Vor-indiziert	A1 12	Der Akku wird mit dem Inhalt der Zelle geladen, deren Adresse in der Zelle steht, deren Adresse sich aus der Summe von 0012 und dem X-Register ergibt (schrecklich kompliziert, nicht wahr?)
Relativ	F0 12	Wenn das Null-Flag (Z) gesetzt ist, erfolgt ein Sprung um 12 Byte nach vorn
Zero Page, indiziert	B9 40	Der Akku wird mit dem Inhalt der Zelle geladen, deren Adresse sich aus der Summe von 0040 und dem Inhalt des X-Registers ergibt

Wie man leicht erkennt, erlaubt die äußerst flexible Adressierung des MCS 6502 die Verwendung vieler Befehle, die nur zwei Byte lang sind. Dabei wird die Zero-Page-Adressierung verwendet, wobei das höherwertige Adressenbyte gleich Null gesetzt wird. Dem Anwender stehen damit praktisch 256 Zellen zur Verfügung, die mit nur zwei Bytes erreicht werden können.

Für die Speicherung der Rücksprung-Adressen bei Unterprogramm- oder Interrupt-Operationen wird normalerweise ein „Stack" verwendet, der beim MCS 6502 fest auf den Speicherbereich 0100...01FF programmiert ist und nicht verändert werden kann. Die Stacklänge beträgt also 256 Byte, so daß maximal 128 Rücksprungadressen gespeichert werden können.

Es existieren drei Interrupt-Ebenen: Reset, NMI und IRQ.

Reset ist ein nicht maskierbarer Interrupt mit der höchsten Priorität. NMI ist ebenfalls ein „nicht maskierbarer Interrupt", während IRQ „Interrupt Request" bedeutet und nur dann tatsächlich bedient wird, wenn das Interrupt-Disable-Bit im Statusregister auf log. 0 liegt. Alle Interrupt-Arten führen zu einem Sprung an bestimmte Interrupt-Adressen, die vom Anwender in gewissen Zellen (FFF...) frei programmiert werden können. Der Interrupt IRQ ist auch innerhalb eines Programms durch den Break-Befehl zu simulieren (Hexadezimal 00). NMI und IRQ führen zum Abspeichern der Rücksprungadressen und des Statusregisters im Stackbereich.

Da die CPU MCS 6502 zu den preiswertesten und dennoch zu den leistungsfähigsten gehört, fand sie weite Verbreitung im Hobbycomputer-Bereich; so enthalten z. B. die Systeme KIM-1, VIM, PET 2001 und Apple II alle diese CPU.

Fe.

Z80

Die CPU Z80 von Zilog zählt gegenwärtig zu den komfortabelsten und schnellsten 8-bit-Prozessoren der dritten Generation. Sie ist, besonders was die Software anbelangt, stark an das bekannte 8080-System angelehnt, darüber hinaus aber wesentlich erweitert.

In MOS-LSI-Technologie gefertigt und im 40 Pin-DIL-Gehäuse verpackt, ist der Z80 so konzipiert, daß (neben den maßgeschneiderten PIO-, CTC-, DMA-Bausteinen) beinahe alle gängigen Speicher (dynamisch, statisch) und Ein/Ausgabe-Bausteine für ein System verwendet werden können. Programme, die auf einem 8080 laufen, laufen auch auf dem Z80 – natürlich nicht umgekehrt, es sei denn, man ließe die zusätzlichen 78 Befehle ungenutzt. Die schnellere Version, der Z80 A (4-MHz-Takt gegenüber 2 MHz) kann den schnellen Prozessoren (z. B. 6502) durchaus das Wasser reichen. Er ist außer den dynamischen Daten völlig identisch mit seinem Bruder.

Der Registersatz

Das *Bild* zeigt die Register des Z80. Bemerkenswert ist, daß die vom 8080 her bekannten Register hier doppelt ausgeführt sind; man kann per Programm zwischen beiden Sätzen umschalten (es ist aber immer nur ein Satz direkt ansprechbar), was für schnelle Interruptbehandlung ideal ist. Außerdem existieren Index-Register für indizierte Adressierungen, ein Interrupt-Register, das einen Teil der Interrupt-Sprungadresse aufnimmt, sowie ein Refresh-Register. Das Statusregister ist im wesentlichen das gleiche wie beim 8080 bis auf ein Subtractflag (N).

Hardware

Der Z80 benötigt nur eine einzige 5-V-Versorgungsspannung und einen Einphasentakt, also lediglich ein Rechtecksignal, dessen Frequenz übrigens zu Testzwecken drastisch bis fast auf Null verringert werden kann.

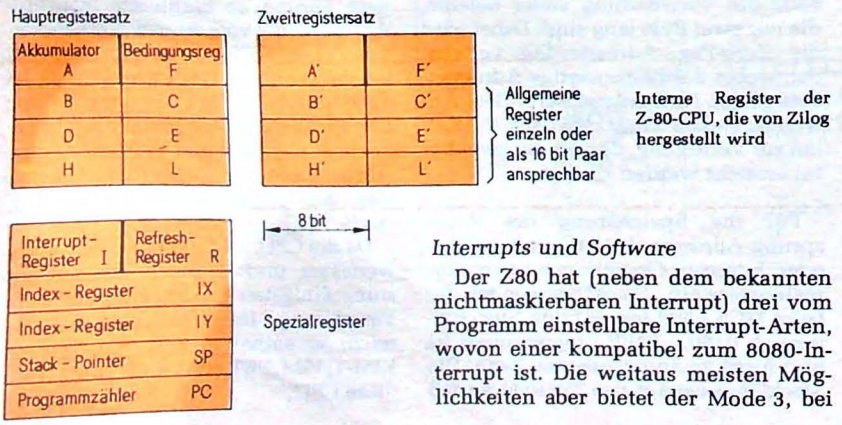
Allgemeine $\overline{RD}$- *und* $\overline{WR}$-*Ausgänge* in Verbindung mit $\overline{MREQ}$ (Memory Request)- bzw. $\overline{IORQ}$ (In/Out Request)-Auswahlausgängen sind günstigerweise gegenüber den separaten Auswahlausgängen des 8080 eingeführt.

Der $\overline{M1}$ (Maschinenzyklus)-Ausgang sagt aus, daß die CPU sich gerade einen Befehlscode aus dem Speicher holen will (Programmzählerinhalt liegt am Adreßbus). Er ist nicht identisch, aber ähnlich dem SYNCH-Impuls seines Vorfahren 8080.

Der $\overline{HALT}$-Ausgang hat nichts mit dem HOLD des 8080 zu tun, er wird durch einen Software-Befehl erreicht. Es werden dann solange Leerbefehle ausgeführt (um Refresh zu garantieren), bis die CPU durch einen Interrupt oder Reset wieder aus dem Schlaf gerissen wird. Dem HOLD-Zustand entspricht vielmehr der $\overline{BUSRQ}$ (Bus Request)- und als Quittung der $\overline{BUSAK}$ (Bus Acknowledge)-Ausgang, bei dem sämtliche Bus- und Steuerausgänge in den hochohmigen Zustand versetzt werden, um direkten Speicherzugriff zu ermöglichen.

Der eingebaute Refresh-Zyklus ist erfrischend sowohl für Hardware-Planer als auch für dynamische Speicher. Am Ende jedes „OP-Fetch"-Zyklus wird, während die CPU ohnehin intern mit der Befehlsverarbeitung beschäftigt ist, (durch $\overline{RFSH}$ und $\overline{MREQ}$ angekündigt) eine Adresse aus dem Refresh-Register angeboten, die direkt für Auffrischzugriffe verwendet werden kann.

Die Zustände während eines jeden Maschinenzyklus können durch ein Signal am $\overline{WAIT}$-Eingang praktisch beliebig verlängert werden, um phlegmatischeren Peripheriebausteinen das Mitreden zu erleichtern.



Interrupts und Software

Der Z80 hat (neben dem bekannten nichtmaskierbaren Interrupt) drei vom Programm einstellbare Interrupt-Arten, wovon einer kompatibel zum 8080-Interrupt ist. Die weitaus meisten Möglichkeiten aber bietet der Mode 3, bei dem ein 8-bit-Interruptvektor vom anfordernden I/O-Port zusammen mit dem Inhalt des Interrupt-Registers praktisch jeden beliebigen Platz im 64-K-Bereich anzuspringen gestattet.

Zur Adressierung ist zu sagen, daß beinahe alle üblichen Arten und deren Mixturen möglich sind: direkte, register-indirekte, indizierte (über die beiden Index-Register), relative und sogar Einzelbit-Adressierung; selbstverständlich auch implizit, z. B. Unterprogrammaufruf für 8 Stellen im Zero-Page-Bereich mit einem 1-Byte-Befehl.

Der Befehlssatz umfaßt:

8-bit-Ladebefehle: Alle Adressierungsarten und Konstantenoperationen.

16-bit-Ladebefehle: Hier besitzt die CPU etwas weniger Adressierungsmöglichkeiten.

Blocktransfer- und Suchbefehle: Der Z80 ist als einziger Prozessor in der Lage, bis zu 64 kByte mit einem einzigen Befehl (Übertragungsraten von ca. 5 µs/byte) von einem Block in den anderen zu übertragen oder ein Zeichen zu suchen.

8-bit-Arithmetisch/logische Befehle: Konstantenaddition, Addition und Subtraktion mit und ohne Carry, AND, OR, XOR und Vergleiche.

16-bit-Arithmetik: Addition, Subtraktion, Inkrementieren und Dekrementieren.

Rotations- und Schiebebefehle: Neben den normalen Rotations- und Shift-Befehlen sind auch Halbbyte-Swaps möglich – eine wesentliche Erleichterung bei BCD-Arithmetik.

Einzelbitverarbeitung: Setzen und Rücksetzen bzw. Testen von Bits in Registern oder indirekt oder indiziert adressierten Speichern.

Sprungbefehle: Bedingte und unbedingte (auch relative) Sprünge und Unterprogrammaufrufe, sowie Ein-Byte-Calls im Zero-Page-Bereich.

Ein/Ausgabe-Befehle: Direkt oder registerindirekt adressierbare Ports (letztere können ihre Daten in jedes beliebige Register laden).

CPU-Steuerbefehle: Interruptmaskierung und Modusfestlegung, HALT und NOP-Befehl.

Karl Schedler

Das nachfolgend abgedruckte Programm für den Rechner TI-59 soll das Herausfinden von Nebenempfangsstellen bei Superhet-Empfängern vereinfachen, um schon vor dem Bau eines solchen Geräts systembedingte Pfeifstellen zu vermeiden.

Berechnung von Nebenempfangsstellen mit einem TI-59

Das Programm verwendet die Formel

$$f_{stör} = \frac{f_{Zf} \pm n \cdot f_{osz}}{m}$$

und berücksichtigt somit die Tatsache, daß sowohl die störende Eingangsfrequenz als auch die intern im Empfänger erzeugte Oszillatorfrequenz Harmonische aufweisen. Diese Harmonischen entstehen u.U. auch erst durch eine unlineare Kennlinie des Mischers und können ebenso zur Bildung der Zwischenfrequenz (Zf) führen wie die Grundwellen, und zwar entweder durch Addition oder Subtraktion.

Um nicht unnötig viele, zum Teil gar nicht interessierende Störfrequenzen zu erhalten, gestattet das Programm auch die vorherige Eingabe zweier Frequenzgrenzen f_u und f_o. Nur innerhalb dieser Grenzen werden Nebenempfangsstellen angezeigt; alle anderen werden ignoriert.

Um feststellen zu können, die wievielte Harmonische von Eingangs- und Oszillatorfrequenz an der jeweiligen Nebenempfangsstelle beteiligt war, werden n und m in Form der letzten beiden Stellen der angezeigten Störfrequenz ausgegeben; die Anzeige 145,000021 bedeutet z. B., daß die Nebenempfangsstelle durch die doppelte Oszillatorfrequenz und die Grundwelle der Eingangsfrequenz zustandekommt. Die Störfrequenz ist hier natürlich nicht 145,000021 MHz, sondern 145,000 MHz; das Programm schneidet nach der dritten Stelle hinter dem Komma ab.

Das *Bild* zeigt das komplett mit einem Drucker PC-100A aufgelistete Programm, und aus der *Tabelle* geht seine Bedienung hervor. Anzumerken ist noch, daß alle Störfrequenzen ab Register 10 gespeichert werden. Die Speicher, auch die vorher eingegebenen Werte, werden durch das Programm gelöscht.

Ing. (grad.) Fred-H. Heller

Tabelle: Bedienung des Rechners

Schritt	Bedienung	Eingabe	Taste	Anzeige
1	Magnetkarte einlesen			1
2	Zwischenfrequenz eingeben	Zf	A	Zf
3	Oszillatorfreq. eingeben	f_{osz}	B	f_{osz}
4	Untere Grenzfrequenz	f_u	C	f_u
5	Obere Grenzfrequenz	f_o	D	f_o
6	Max. Oszil.-Harmonische	n_{max}	E	n_{max}
7	Max. Eingangs-Harmonische	m_{max}	E' R/S	m_{max}; 91

(Die Nebenempfangsstellen werden jetzt berechnet und gespeichert.)

8	Abwarten, bis 91 angezeigt wird		R/S	0
9	Ausgabe von $f_{stör}$		R/S	f,fff0nm
10	R/S drücken bis 91 (STOP) angezeigt wird		R/S usw.	
11	Wieder bei Schritt 2 beginnen		R/S	91

```
000  76 LBL    034  39  39    068  71 SBR    102  65  ×     136  01  1     170  55  ÷
001  11  A     035  76 LBL    069  67  EQ    103  01  1     137  95  =     171  43 RCL
002  42 STO    036  10 E'     070  71 SBR    104  00  0     138  72 ST*     172  05  05
003  00  00    037  42 STO    071  75  -     105  00  0     139  06  06    173  95  =
004  61 GTO    038  08  08    072  71 SBR    106  00  0     140  92 RTN     174  50 I×I
005  00  00    039  91 R/S    073  67  EQ    107  95  =     141  76 LBL     175  92 RTN
006  39  39    040  09  9     074  43 RCL    108  59 INT    142  85  +     176  09  9
007  76 LBL    041  42 STO    075  08  08    109  55  ÷     143  53  (     177  01  1
008  12  B     042  06  06    076  32 X:T    110  01  1     144  43 RCL     178  91 R/S
009  42 STO    043  43 RCL    077  43 RCL    111  00  0     145  00  00    179  25 CLR
010  01  01    044  07  07    078  05  05    112  00  0     146  85  +     180  32 X:T
011  61 GTO    045  32 X:T    079  67  EQ    113  00  0     147  43 RCL     181  01  1
012  00  00    046  43 RCL    080  00  00    114  85  +     148  04  04    182  00  0
013  39  39    047  04  04    081  43  43    115  43 RCL    149  65  ×     183  42 STO
014  76 LBL    048  67  EQ    082  61 GTO    116  04  04    150  43 RCL     184  09  09
015  13  C     049  01  01    083  00  00    117  65  ×     151  01  01    185  25 CLR
016  42 STO    050  76  76    084  64  64    118  00  0     152  54  )     186  91 R/S
017  02  02    051  69 OP     085  76 LBL    119  93  .     153  55  ÷     187  73 RC*
018  61 GTO    052  24  24    086  67  EQ    120  00  0     154  43 RCL     188  09  09
019  00  00    053  01  1     087  32 X:T    121  00  0     155  05  05    189  69 OP
020  39  39    054  42 STO    088  43 RCL    122  00  0     156  95  =     190  29  29
021  76 LBL    055  05  05    089  02  02    123  00  0     157  92 RTN     191  22 INV
022  14  D     056  71 SBR    090  77  GE    124  01  1     158  76 LBL     192  67  EQ
023  42 STO    057  85  +     091  01  01    125  85  +     159  75  -     193  01  01
024  03  03    058  71 SBR    092  40  40    126  43 RCL    160  53  (     194  86  86
025  61 GTO    059  67  EQ    093  43 RCL    127  05  05    161  43 RCL     195  47 CMS
026  00  00    060  71 SBR    094  03  03    128  65  ×     162  00  00    196  09  9
027  39  39    061  75  -     095  22 INV    129  00  0     163  75  -     197  01  1
028  76 LBL    062  71 SBR    096  77  GE    130  93  .     164  43 RCL     198  61 GTO
029  15  E     063  67  EQ    097  01  01    131  00  0     165  04  04    199  00  00
030  42 STO    064  69 OP     098  40  40    132  00  0     166  65  ×     200  39  39
031  07  07    065  25  25    099  32 X:T    133  00  0     167  43 RCL
032  61 GTO    066  71 SBR    100  69 OP     134  00  0     168  01  01
033  00  00    067  85  +     101  26  26    135  00  0     169  54  )
```

Programm zur Berechnung der Nebenempfangsstellen. Es umfaßt genau 200 Schritte

Ing. (grad.)
Wolfgang Hascher

Das Einarbeiten in die Mikrocomputertechnik und – was mindestens genauso wichtig ist – in die Programmierpraxis bedarf eines wohldurchdachten µC-Systems, denn zunächst soll das Lernen und Üben frei sein von unnötigem Ballast. Weiterhin bestimmt wohl der Mikroprozessortyp, mit dem man begonnen hat, die Richtung des späteren Arbeitens auf dem Computersektor: Hard- und auch Software wird man kaum mehr wechseln wollen.

Näher beleuchtet: µC-System Alpha 1

Insofern stellen sich dem Übenden, dies sei vorweggenommen, mit dem hier vorliegenden Mikrocomputersystem Alpha 1 der Berliner Firma MCS schon recht gute Voraussetzungen für ein erfolgreiches Arbeiten: Der Mikroprozessor 6502 (CPU) ist allseits anerkannt und bietet einen in der Praxis bestens bewährten Befehlssatz. Um ihn herum wurde sinnvolle Peripherie aufgebaut, so daß das System mit den Erfahrungen und Bedürfnissen „mitwachsen" kann und mit seinen Erweiterungsmöglichkeiten zahlreichen praxisbezogenen Anwendungen gerecht wird. Letztlich können sogar anspruchsvolle und komplizierte Steuerungsfunktionen realisiert werden!

Allgemeines und Hardware

Der Alpha 1 ist ein anschlußfertiges 8-bit-Mikrocomputersystem (1,25 K RAM, 2 K Betriebssystem) mit eingebautem Netzteil, lieferbar in drei Versionen: Typ 1 (*Bild*) mit der charakteristischen „aufgesetzten" CPU-Karte zum Preis von knapp 1500 DM + MwSt. Der Typ 2 kostet etwas über DM 1800.– zuzügl. MwSt. und hat die CPU-Karte bereits im Gehäuse. Die letzte Version, der Typ 3, befindet sich im durchsichtigen Kunstglasgehäuse, kostet knapp über 1900 DM + MwSt. und läßt sich wiederum „auf den Prozessor schauen". Als Bausatz ist lediglich der erste Typ lieferbar: Wißbegierige und schon im Löten erfahrene Heimcomputerleute haben hierfür knapp unter 1000 DM + Steuer aufzubringen.

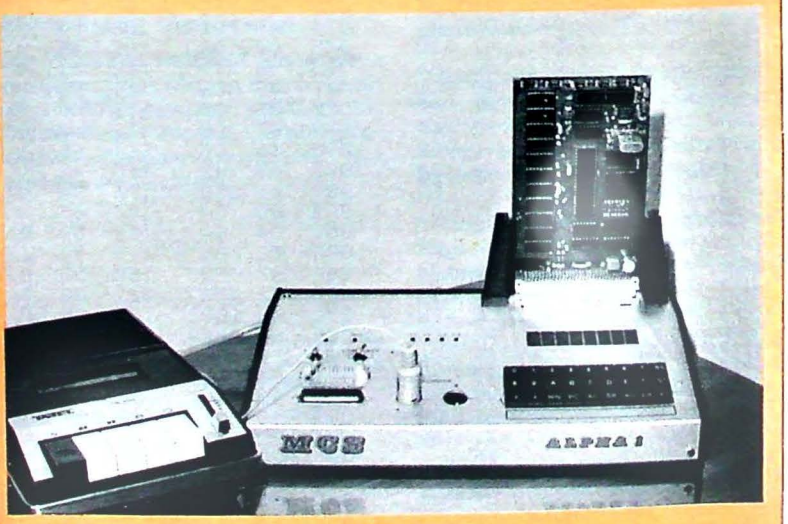
Nun zur Bedienung: Auf eine Handhabung, die das Programmieren erleichtert und nicht ermüdend gestaltet, ist die 27 Elemente umfassende Tastatur zugeschnitten, mit deren Erläuterung auch gleich wesentliche Eigenschaften des gesamten Systems deutlich werden: Neben den 16 Hexadezimaltasten (0...9 und A...F) sind vorhanden – und werden beim Programmerstellen auch häufig benötigt:

- eine Reset-Taste, die einen definierten Anfangszustand herstellt;
- Einzelschritt-Programmtaste, besonders wichtig beim Austesten von Zeitschleifen und Programmsprüngen;
- „Go", startet den Programmzyklus bei der momentan angewählten Adresse;
- Tasten „AD" und „DA", mit deren Hilfe man über die Hexadezimaltastatur wahlweise Zugriff zu Adressen (Speicherplätzen) und den darin stehenden Daten bekommt;
- Taste „SR"; sie läßt den Inhalt des Statusregisters in der Anzeige erscheinen;
- „AC", eine Funktion, die den Dateninhalt des Akkumulators auf das Display bringt – sehr vorteilhaft, da man nicht ständig die vierstellige Akku-Adresse eintasten muß;
- die Taste „PC" erlaubt die Direktanwahl des Programmzählers;
- „MN", eine Funktion von besonderer Bedeutung: Mit ihrer Hilfe nämlich kann jeder hexadezimale Befehlscode gleich in mnemonischer Darstellung abgelesen werden (Disassembler), also statt z. B. A 9 erscheint gleich LDA (load accumulator), was eine große Hilfe besonders für den noch Ungeübten bedeutet;
- schließlich noch zwei Tasten mit Pfeilsymbolen (↑, ↓). Sie zählen die aktuelle Adresse um Eins nach oben bzw. unten, ersparen also beim Programmieren das Neueingeben der nächsten oder letzten Adresse – eine erhebliche Zeitersparnis.

Was die schon erwähnte Anzeige betrifft, so hat sie lediglich einen Mangel: ihr fehlt ein rotes Farbfilter, das die Siebensegmentsymbole angenehmer für das Auge darstellen würde. Ansonsten sind die acht Stellen gut abzulesen; auf ihnen erscheinen im Normalbetrieb die Adresse des Speicherplatzes und die darin vorhandenen Daten gleichzeitig.

Peripherie

An externen Anschlußmöglichkeiten bietet das System zwei Recorderbuch-



Das µC-System Alpha 1 (Typ 1) mit „aufgesetzter" CPU-Karte, die bei Systemerweiterung entnommen werden kann; die Anschlüsse dienen dann als Busleitungen. Mit einem Billig-Kassettenrecorder lassen sich alle Programme abspeichern und wieder einlesen

sen, mit denen sowohl Ein/Auslesen als auch Überspielen von (gleichzeitig regenerierten) Daten auf ein zweites Bandgerät möglich ist; die Laufwerke der Kassettengeräte werden über einen Relaiskontakt mitgesteuert.

Am „Application"-Stecker (20polig) liegen die 16 frei adressierbaren Ein- bzw. Ausgangsleitungen des Peripheriebausteins 6532-1; sie dienen den frei verfügbaren Eingabe-, Kontroll- und Steuerfunktionen. Viele Möglichkeiten bietet auch der „Communication"-Stecker. Er erlaubt an einer 20-mA-Stromschnittstelle den Betrieb eines Teletype inkl. Lochstreifen. Für Datenterminals steht eine V-24-/RS-232-Schnittstelle zur Verfügung, insgesamt mit Übertragungsgeschwindigkeiten bis 4800 Baud.

Software

Die bisher geschilderten Möglichkeiten und die Ausstattung des Alpha-1-Systems deuteten es schon an: hier muß ein ausgefeiltes Betriebssystem (Monitorprogramm) am Werk sein, um die Hardware und die Peripherie voll auszunutzen. Es wurde auf 2 K untergebracht und erlaubt vollen Zugriff auf Unterprogramme, wie zum Beispiel die in eigenen Programmen immer wieder benötigte Anzeigeroutine. Auch sind alle Tastenfunktionen frei verfügbar, um während des Programmlaufes beliebige Informationen eingeben zu können. Weitere Unterprogramme beziehen sich auf Uhr- und Timeranwendungen, Schreib- und Leseroutinen für alle Peripheriegeräte, Magnetbandaufzeichnung und -wiedergabe; sogar bei der Ausgabe von Tonfolgen ist das Betriebssystem behilflich.

Natürlich wäre die ganze Software nutzlos ohne ausreichende Dokumentation und Erläuterung; dieser Gefahr wurde jedoch vorgebeugt durch gründliche Beschreibungen, Anwendungsbeispiele und Programmlisten mit Sprungadressen – man kann in der Tat das System auspacken, anschließen und gleich zu programmieren beginnen; erste Erfolgserlebnisse (sie sind besonders wichtig) stellen sich bald ein.

Auch der Software-„Nachschub" wird bei MCS recht ernst genommen. Erhältlich sind Programmkassetten mit Test- und Prüfdurchläufen für das gesamte System und die Peripherie sowie interessante Bänder mit Unterhaltungs- und Spielprogrammen (u. a. Schach, Mondlandung, Musikbox). Lieferbar sind zudem Programmierhilfen wie Assembler, Disassembler, Debug-Programm, ASCII-Erkennung und, neben weiteren, ein komfortabler Editor. Im Software-Katalog nicht unerwähnt bleiben schließlich Ergänzungsprogramme für die Ansteuerung von Floppy-Disks, Druckern und Terminals mit verschiedensten Baudraten.

Alles in allem kann das Mikrocomputersystem Alpha 1 als ausgereiftes Produkt gesehen werden, mit einem weitverbreiteten Prozessor und – die CPU sinnvoll einsetzend – einem recht praxisorientierten und nützlichen Monitorprogramm. Steigender Programmiererfahrung und (je nach Geldbeutel) erweiterbarer Peripherie kann sich das System optimal anpassen, zumal auch entsprechende Dokumentation und Software verfügbar sind. Letztlich also: gutes Geld für gute Ware.

KIMpatibel

Die Tatsache, daß selbst Mikrocomputer-Systeme mit dem gleichen Prozessor nicht unbedingt software-kompatibel sind, liegt meist daran, daß unterschiedliche RAM-Speicheradressen verwendet werden.

Bei dem Mikrocomputer KIM-1 sind die verbreitetsten Erweiterungen die RAM-Adressen 0400...13FF oder 2000...2FFF. Beide Erweiterungsmöglichkeiten lassen sich mit einer schon recht preiswerten 4-kByte-RAM-Karte realisieren. Programme, die für den Adressenbereich 0400...13FF geschrieben wurden, laufen nun aber leider nicht ohne weiteres im Bereich 2000...2FFF, es sei denn, man ändert alle absoluten Adressen im Programm, was recht mühsam ist.

In der FUNKSCHAU werden Sie in dieser Rubrik jedoch auch solche Programme finden, die eine der beiden Erweiterungen voraussetzen, weil sie sich nicht in dem auf der KIM-Platine vorhandenen RAM im Adressenbereich 0000...03FF unterbringen lassen. Um Ihnen nun die Arbeit mühsamer Programmverschiebungen zu ersparen, sei eine Möglichkeit aufgezeigt, die dieses Problem löst. Auf ähnliche Art und Weise läßt sich das natürlich auch bei anderen Systemen durchführen.

Das *Bild* zeigt die Schaltung der Hardware-Adressenduplizierung. Es ist keine manuelle Umschaltung erforderlich, in welchem Adressenbereich gearbeitet werden soll – das tut das Programm selbst durch Ansprechen der Adressenleitung AB 13. Ist diese Leitung H, so ist der Bereich 2000...2FFF gemeint, andernfalls der Bereich 0400...13FF.

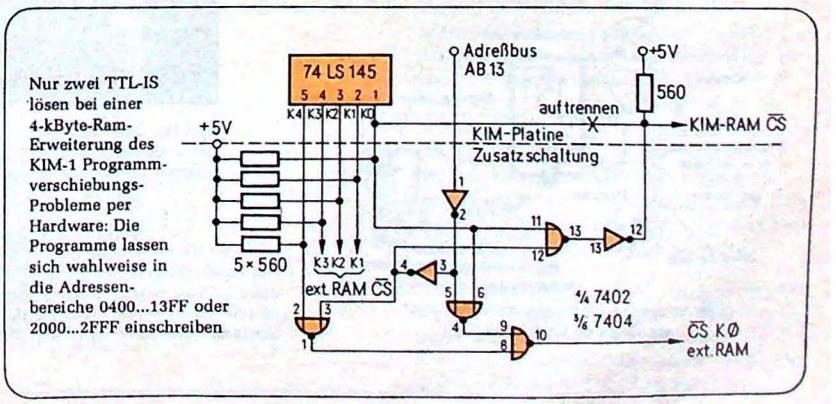
Je nach dem Zustand von AB 13 liegt das externe RAM also in folgenden Adressenbereichen:

2000...23FF oder 1000...13FF und
2400...2FFF oder 0400...0FFF.

Der zweite Bereich, der 3 kByte umfaßt, ist fest mit dem auf der KIM-Platine befindlichen Adreßdecoder verbunden. Lediglich der erste Bereich – er umfaßt 1 kByte – wird, abhängig von AB 13, mit der aus nur zwei TTL-IS bestehenden Logik als K 0 oder K 4 interpretiert.

Das KIM-RAM wird gesperrt, solange AB 13 auf H liegt. Dazu muß die Leiterbahn auf der KIM-Platine zwischen Pin 1 des Adreßdecoders 74 LS 145 und der Chip-Select-Leitung des KIM-RAM aufgetrennt werden, was mit einem scharfen und spitzen Messer geschehen kann.

Diese Hardware-Programmverschiebung läßt sich natürlich nur dann durchführen, wenn der höherwertige Adressenteil, also AB 13...15, nicht von der externen RAM-Platine selbst decodiert wird. Fe.



Hans Werner

Mikrocomputer

Eine kleine Einführung

4. Teil

Schon fast vollständig...

...ist der Computer, wenn Schalter und Startknopf durch entsprechende Register und Zähler ersetzt werden:

Die drei Befehlsschalter werden durch das Befehlsregister BR, die fünf Adreßschalter durch das Datenregister DR ersetzt.

An Stelle des Startknopfes, der vor jeder Befehlsausführung erneut bestätigt werden mußte, tritt der Programmzähler PZ. Er ist ein einfacher Binärzähler, der beim Programmstart die Stellung 0 hat und nach jedem vollendeten Befehl durch einen Impuls am Eingang CL um 1 erhöht wird. Damit ändert sich die Speicheradresse in der Reihenfolge 0, 1, 2...

Die Adresse wird aber nur dann ausgegeben, wenn der Steuereingang PZ, A = 1 ist. Das Programm ist im Speicher in der gleichen Reihenfolge abgelegt und wird nacheinander abgearbeitet.

Beispiel:
In Speicherzelle 0 steht

100 10100	LA 20

In Speicherzelle 1 steht

000 10101	AD 21

Nach dem Start gibt der PZ die Adresse 0 heraus, der Inhalt der Speicherzelle wird in den Akku geladen und der Programmzähler um 1 erhöht usw. Der genaue Ablauf kann nur mit dem Befehlszyklus 1...3 erklärt werden, der noch um weitere Steuerbefehle zu vervollständigen ist.

Der Befehlszyklus ist aus den Intervallen 1...3 zusammengesetzt, die durch extern erzeugte Taktimpulse in dem Steuerteil St entstehen. Das Steuerteil kann z. B. ein dreistufiges Schieberegister sein, dessen Ausgänge direkt bzw. über ODER-Verknüpfungen auf die Impulsausgänge geführt werden.

Intervall 1: Ausgabe von PZ und SP; Eingabe BR, DR.

Intervall 2: Ausgabe von Dr, BR, AC, SP; Eingabe von ZS.

Intervall 3: Ausgabe von DR und ZS; Eingabe von AC. Clockimpuls CL an PZ; mit Impuls CL wird der Zählerstand um 1 erhöht.

Steuerteil und Programmzähler sollen beim Starten des Programmes aus einer definierten Grundstellung zu arbeiten beginnen und am Programmende anhalten.

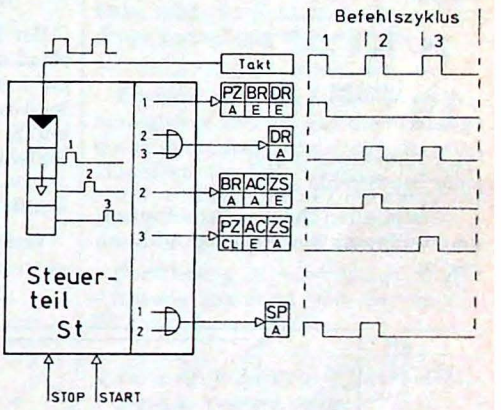
Start

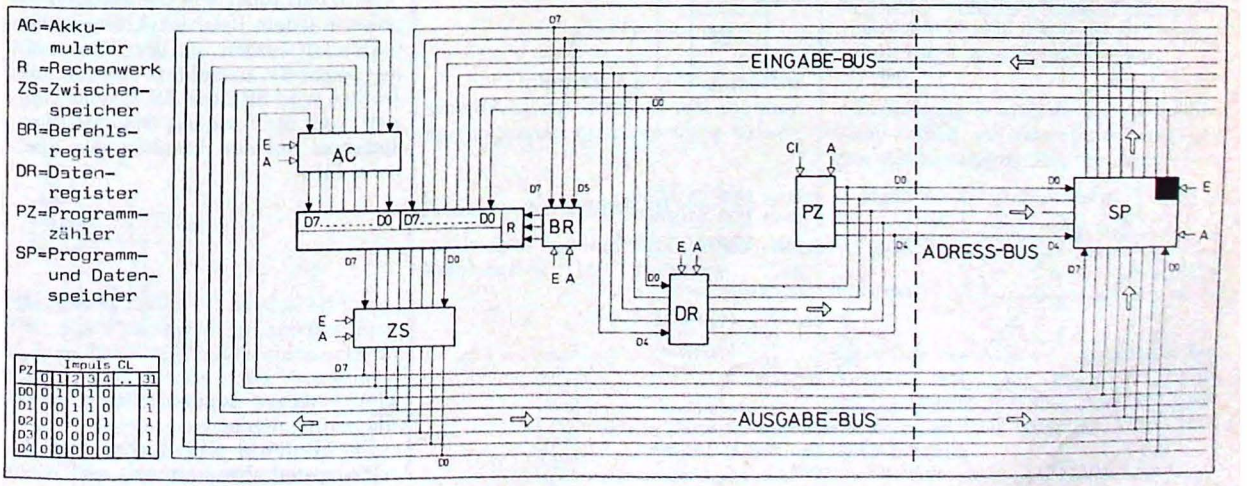
Durch eine gesonderte Taste oder durch Einschalten der Stromversorgung wird: a) der Programmzähler auf 0 gesetzt; b) das Steuerteil in die Anfangsstellung „Intervall 1" gebracht.

> Der Startbefehl setzt Steuerteil und Programmzähler auf 0. Er ist nicht im Programm enthalten.

Stop

Am Ende jedes Programmes ist der Stop-Befehl 111 erforderlich, der alle Impulsreihen des Steuerteiles unter-



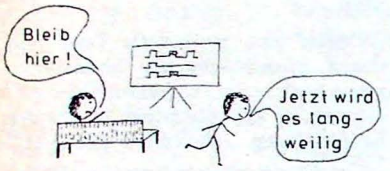


PZ	Impuls CL						
	0	1	2	3	4	..	31
D0	0	1	0	1	0		1
D1	0	0	1	1	0		1
D2	0	0	0	0	1		1
D3	0	0	0	0	0		1
D4	0	0	0	0	0		1

drückt. Er kann nur durch einen erneuten Start aufgehoben werden.

> Der Stop-Befehl steht am Ende des Programmes und verhindert ein unnötiges Weiterlaufen des PZ.

(Der Start- bzw. Nullstelleingang und die Dekodierung des Stop-Befehls sind nicht im Schaltbild eingetragen.)



Befehlsabläufe

Es ist nicht sinnvoll, die Befehlsabläufe eines Modellcomputers zu erlernen.
An ihnen sollen lediglich prinzipielle Möglichkeiten gezeigt und erkannt werden.

Die arithmetischen und logischen Befehle...

...haben alle den gleichen Befehlsablauf und werden am Beispiel einer Addition erklärt.

Start

Programmzähler und Steuerteil auf Null.

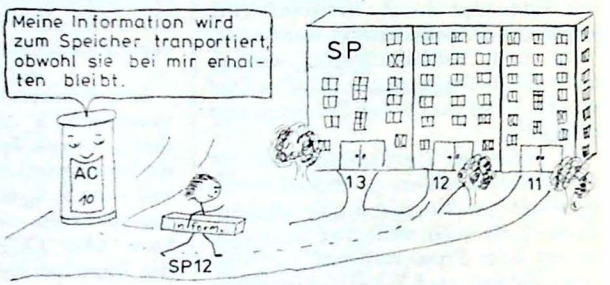
Intervall 1

Der PZ bekommt seinen Ausgabebefehl und spricht Speicherzelle 0 an.

Nach den E-A-Befehlen für Speicher und Register steht im BR der Befehl ADDIERE (000) und im DR die Adresse 12 (01100). Daß der Inhalt von Zelle 12 auch zum Rechenwerk gelangt, ist ohne Bedeutung.

Intervall 2 und 3

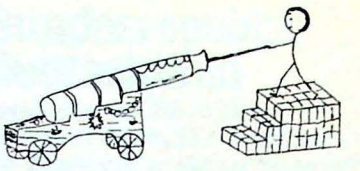
Das Datenregister DR adressiert Zelle 12, deren Inhalt zum Rechenwerk gelangt. Mit BR, A = 1 erhält das Rechenwerk den Befehl AD. Danach spielt sich der schon bekannte Vorgang „Rechnen und Umspeichern in den Akku" ab.



Der Impuls PZ, CL erhöht den Zustand des Programmzählers um 1, also auf 1. Im Intervall 3 tritt noch ein zweiter Ausgabebefehl für das Datenregister auf. Er hat bei den arithmetischen und logischen Befehlen keine Bedeutung.

Der Befehl LADEN

verläuft, bis auf das Verhalten des Rechenwerkes, vollkommen identisch ab.

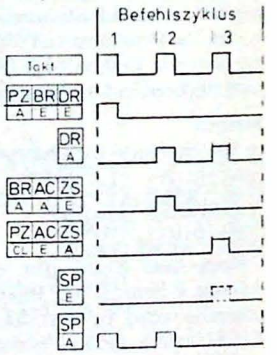


Der rechte Eingang des Rechenwerkes wird durchgeschaltet und der Inhalt der gewünschten Speicherzelle in den Akkumulator geschoben.

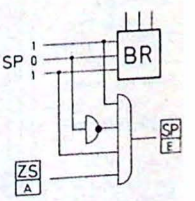
Der Befehl SPEICHERN

ist mit der bisherigen Zyklussteuerung zunächst nicht möglich. Es fehlt der Steuerimpuls SP,E (Eingabe Speicher).

Um SP,E erweitertes Diagramm:

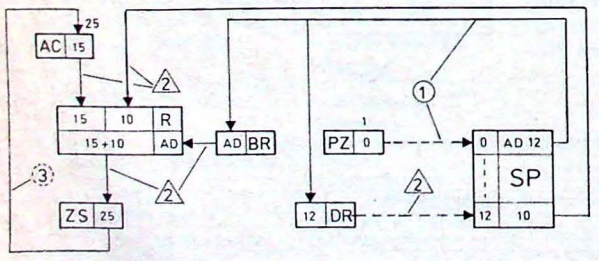


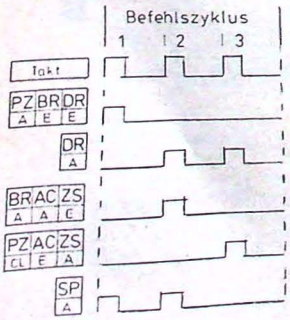
SP,E darf nicht wie die anderen Impulse in jedem Befehlszyklus automatisch zu 1 werden, sondern nur wenn der Befehl SP ausgeführt werden soll. Zeitlich wird SP,E mit ZS,A synchronisiert, weil der Ausgang des Zwischenspeichers auf den Eingang des Speichers führt.



Eine einfache Logikschaltung erfüllt die Bedingungen.

Der SP-Befehl wird hinter dem Befehlsregister abgenommen und steht





In Zelle 0 steht: 000 01100
In 12 steht: 000 01010
Im Akku steht: 000 01111

In normaler Schreibweise:

0 AD 12 (Befehl)
12 10
AC 15 (Zahlen)

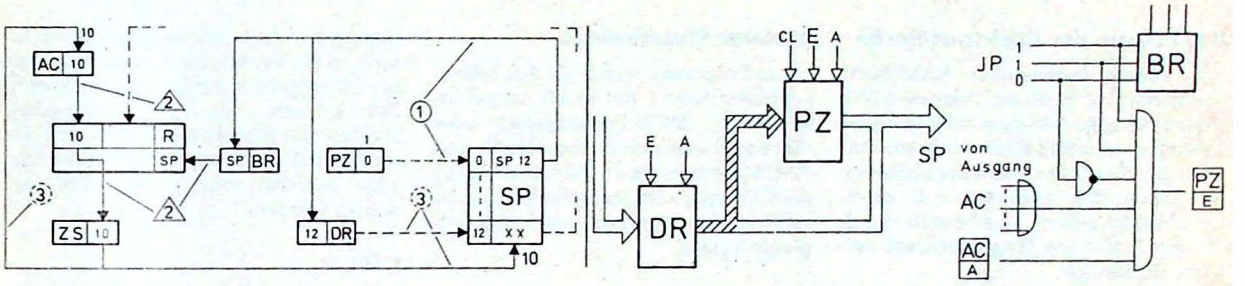
Das bedeutet, der Inhalt der Speicherzelle 12 soll zum Inhalt des Akkumulators addiert werden.

mit Intervall 2 zur Verfügung. Mit Intervall 3 kommt der ZS,A-Impuls und SP,E wird 1.

Beispiel

In Speicherzelle 0 steht SP 12, im Akku die Zahl 10 und in Speicherzelle 12 ein beliebiger Wert XX.
Der Inhalt des Akkus soll mit dem Befehl SP 12 in Speicherzelle 12 geschoben werden.

Inhalt des Akkus ungleich 0, so wird das Programm mit Schritt 13 weitergeführt. Steht im Akku eine 0, läuft das Programm normal ab.



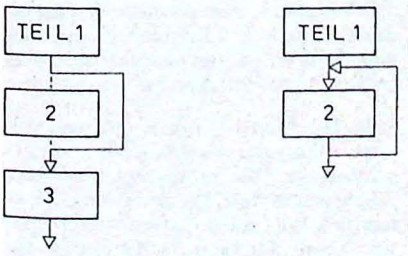
Intervall 1:
Adressieren von Zelle 0 und Umspeichern des Befehls in die Register BR und DR.

Intervall 2:
Der Inhalt von BR und AC liegt am Rechenwerk, das die 10 direkt auf den Eingang des Zwischenspeichers schaltet. (DR,A und SP,A sind ohne Einfluß).

Intervall 3:
Ausgabe des Zwischenspeicher-Inhaltes in Zelle 12 und in den Akku. Erhöhen des Programmzählers um 1.

Sprungbefehl JP

Der Programmzähler beginnt mit Adresse 0 und zählt dann nach jedem ausgeführten Befehl um 1 weiter. Das Programm wird also in der Reihenfolge 0, 1, 2, 3... abgearbeitet, bis der Stop-Befehl ST die Steuerimpulse unterdrückt und das Programm beendet.

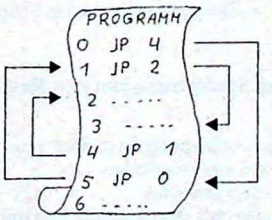


In vielen Fällen ist es aber erforderlich, das Programm zu unterbrechen und an anderer Stelle fortzusetzen. Oder es sollen Teile eines Programms mehrfach durchlaufen werden. Dazu dient der Sprungbefehl JP.
Steht z. B. in Zelle 5 der Befehl JP 12 und ist bei der Befehlsausführung der Inhalt des Akkus ungleich 0, so wird das Programm mit Schritt 13 weitergeführt. Steht im Akku eine 0, läuft das Programm normal ab.

Befehl : JP n
Ist der Inhalt des Akkumulators ungleich 0, wird das Programm mit Adresse n+1 fortgesetzt.

Aufgabe

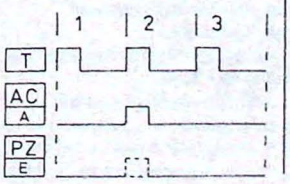
4 Sprungbefehle sind scherzhaft hintereinander angeordnet. Wird Schritt 6 erreicht, wenn im Akkumulator eine 5 steht?



Lösung: Nein, durch die innere Schleife wird für alle Ewigkeit der Programmausschnitt 2, 3, 4 ausgeführt.

Befehlszyklus SPRINGEN

Für den Sprungbefehl wurde die Kombination 110 festgelegt. Das Sprungziel ist hinter dem Befehlsteil angegeben und hat wie bei allen anderen Befehlen eine Länge von 5 Bit. Nach Intervall 1 steht grundsätzlich der Befehlsteil im Befehlsregister und die Adresse im Datenregister.



Wenn nun der Programmzähler mit dem Inhalt des Datenregisters geladen wird, ist das Problem schon gelöst. Im nächsten Befehlszyklus hat der Programmzähler nicht wie üblich um 1 erhöht, sondern adressiert den Speicher mit der frisch geladenen Adresse und das Programm wird an dieser Stelle fortgesetzt.

Dazu muß die Gesamtschaltung allerdings geringfügig erweitert werden.

Erweiterung 1: Die Ausgänge des Datenregisters werden auf die Eingänge des Programmzählers geführt, der noch zusätzlich den Steuereingang E erhält. Mit E = 1 übernimmt PZ die Eingangsdaten.

Erweiterung 2: Eine kleine Logikschaltung garantiert, daß der Impuls PZ,E nur dann auftritt

a) wenn der Befehl 110 (JP) am Befehlsregister-Ausgang steht,

b) wenn der Akkumulatorinhalt ungleich 0 ist. Bei 0 soll der Sprungbefehl nicht ausgeführt werden.

Bedingung b wird durch ein ODER-Glied erfüllt, auf das alle 8 Ausgänge des Akkumulators führen. Nur wenn

(AC)	AC,A	JP	PZ,E
≠ 0	1	JA	1
= 0	1	JA	0

der Inhalt des Akkus 0 ist, ist auch der Ausgang des ODER-Gliedes 0. Damit PZ,E während Intervall 2 wirkt, liegt noch zusätzlich Impuls AC,A am UND-Eingang.

Fortsetzung folgt

Nehmen wir an, Sie hätten eine Kundenkartei mit hundert oder mehr Namen, Adressen und Telefonnummern. Wenn Sie jetzt herausfinden wollen, wer Ihrer Kunden z. B. in Frankfurt wohnt, wer mit Vornamen Max heißt oder wer die Telefonnummer mit den drei Vieren am Ende hat, dann kann die Sucharbeit Stunden dauern – es sei denn, Sie bedienen sich des hier vorgestellten Programms, das für den Mikrocomputer KIM-1 entwickelt wurde.

KIM auf Datensuche

Das Prinzip der Stichwort-Suche

In einem bestimmten RAM-Speicherbereich, z. B. ab der Adresse 0200, stehen Namen, Adressen oder sonstige Textdaten in ASCII-Form, d. h. ein Zeichen pro Byte. Die zusammenhängenden Daten, d. h. alles, was z. B. zu einem Namen gehört, sind jeweils durch 0D (Hex), also ein Wagenrücklauf-Zeichen abgegrenzt.

Wird das Programm gestartet, so erscheint ein Fragezeichen am Beginn einer neuen Zeile. Wenn man nun ein Stichwort eingibt, z. B. MAX, gefolgt von der Return-Taste, so beginnt sofort der Mikrocomputer alle gespeicherten Texte nach diesem Stichwort, besser gesagt, nach der gerade eingegebenen ASCII-Zeichenfolge, abzusuchen. Dann werden alle Texte, die das Stichwort enthalten, nacheinander ausgedruckt.

Enthält der Speicher z. B. einen Text „MAX MEIER, FRANKFURT, 061112345", so wird dieser Text ausgegeben, wenn das Such-Stichwort z. B. MAX, FRANKFURT, FRANK oder 0611 lautete. Dadurch ist ein Suchen praktisch nach beliebigen Kriterien möglich.

Tastatur-Steuerbefehle

Das Programm wurde für den Mikrocomputer KIM-1 mit einem seriell arbeitenden ASCII-Fernschreiber oder -Terminal entwickelt. Eine Reihe von ASCII-Zeichen bzw. Terminal-Tasten sind für spezielle Steuerfunktionen reserviert. Im einzelnen sind dies folgende Tasten:

~(n)

Daten vom Band laden; dabei wird das ASCII-Zeichen (n), z. B. A, B, a, b, 1, 2 usw. als Identifikation benutzt, um einen bestimmten Datenblock ausfindig zu machen. Werden Datenblöcke gefunden, die nicht gesucht werden, so wird deren Identifikation mit ausgedruckt.

ESC

Die Escape-Taste dient zum Umschalten in den Eingabe-Modus. Soll vor der Eingabe ein alter Text gelöscht werden, so ist dieser Taste ein Stichwort nachzustellen, das diesen Text eindeutig definiert. Ist die Löschung vollzogen, so wird das Fragezeichen am Beginn der Zeile durch einen Stern ersetzt. (Das funktioniert natürlich nur bei Sichtgeräten, nicht bei Druckern!). Jetzt können neue Texte eingegeben werden, die jeweils mit der Return-Taste beendet werden müssen. Ist die Eingabe beendet, muß wieder ESC gedrückt werden.

CTRL H

Diese Taste dient – wie üblich – als Back Space und setzt den Cursor um eine Schreibstelle rückwärts. Im Eingabemodus wird dabei gleichzeitig der Speicher-Zeiger um Eins erniedrigt, so daß damit Korrekturen möglich sind, wenn man sich einmal vertippt hat.

(n)

Daten mit (n) als Identifikations-Zeichen auf Band speichern; wie beim Laden des Bandes kann (n) ein beliebiges ASCII-Zeichen sein. Das verwendete Kassetten-Format benützt die auf der KIM-Platine verwendete Hardware, ist aber rund 12mal schneller als die KIM-Monitor-Routine zur Bandaufzeichnung. 1 kByte wird in nur etwa 11 s auf Band gespeichert!

>

Soll ein neuer Text eingegeben werden, ohne einen vorhandenen zu löschen, so ist dieses Zeichen statt des Stichwortes nach der ESC-Taste zu drücken. Nach einem weiteren Tastendruck, nämlich „Return", kann der neue Text eingetippt werden, wie dies unter ESC beschrieben ist.

Zu beachten ist dabei, daß ein von zwei Wagenrücklaufzeichen eingeschlossener Text nicht länger als 255 Zeichen sein darf. Wenn einmal versehentlich ESC gedrückt wurde, obwohl gar keine Eingabe beabsichtigt ist, drückt man einfach etwa 10mal die Leertaste und dann Return. Das zu suchende Stichwort darf übrigens nie mehr als 20 Buchstaben oder Ziffern umfassen.

Die *Tabelle* zeigt ein Beispiel für den Umgang mit diesem recht universell verwendbaren Programm. Zu erwäh-

Tabelle: Beispiel für den Umgang mit dem Suchprogramm (r = Return-Taste, e = Escape)

1873 A9 G	Das Steuerprogramm wird von
KIM	der Kassette geladen...
0000 20 G	...und gestartet.
? ~P	Daten mit dem Kennbuchstaben P
	werden von der Kassette geladen.
0D2B	Fertig, Text-Endadresse ist 0D2B.
? INHALT r	„INHALT" ist hier der Suchbegriff.
INHALT: PRAXIS & HOBBY 1978	Aha! Sehen wir weiter.
? UHR r	Gesucht: Uhren aller Art.
MODUL-UHR MIT WECKSCHALTUNG 1/33	
TASCHENRECHNER ALS SCHALTUHR	
UND ZÄHLER 4/157	
DIGITALUHR MIT MN 5316 16/781	
? e MN 5316 r	Halt! Es muß heißen MM 5316!
DIGITALUHR MIT MM 5316 r	Der richtige Text wird eingegeben.
e	Eingabe-Modus verlassen!
? 5316 r	Zur Kontrolle...
DIGITALUHR MIT MM 5316	...sehen wir es uns nochmal an.
? e > r	Ein neuer Text...
STEREO-NACHBRENNER 19/945 r	...soll hinzugefügt werden.
e	Das ist alles!
? ↑P	Der korrigierte Inhalt wird auf die Kassette geladen, Kennbuchstabe ist wieder P.

nen wäre noch, daß das Funktionieren der Kassetten-Lesefunktion auf der rechten Stelle des KIM-Display überwacht werden kann; das anfängliche Synchronisieren und das Erkennen des Datenstarts ist deutlich zu sehen.

Das Programm belegt den Adressenbereich 0000...01F0; die Text-Startadresse ist normalerweise 0200. Wer diese Textadresse in einen anderen Speicherbereich legen möchte, muß die entsprechende Page in die Zellen 000C, 0105 und 015C schreiben (normalerweise 02). Bei der erstmaligen Eingabe eines Textes muß der Speicher wie folgt intitialisiert werden:

0201 0D (Carriage Return)
0202 3E („>")
0203 00 (End-Zeichen)

Das *Bild* zeigt eine hexadezimale Auflistung des Programms, wie sie mit dem im Hobbycomputer-Sonderheft des gleichen Verlages beschriebenen Super-KIM-Monitor zustandekommt. Es dürfte damit keine Schwierigkeiten bereiten, es auf Anhieb laufen zu lassen.

Herwig Feichtinger

```
0000  20 2F 1E
0003  A9 3F
0005  20 A0 1E
0008  20 9E 1E
000B  A9 02
000D  85 F8
000F  A9 00
0011  85 FA
0013  A2 18
0015  95 DE
0017  CA
0018  D0 FB
001A  4C C8 01
001D  C9 1B
001F  D0 07
0021  A9 FF
0023  85 DF
0025  20 5A 1E
0028  C9 0D
002A  F0 07
002C  95 E0
002E  E8
002F  E0 15
0031  D0 F2
0033  20 63 1F
0036  A0 00
0038  B1 FA
003A  F0 C4
003C  C9 0D
003E  D0 F3
0040  A2 00
0042  C8
0043  B5 E0
0045  F0 0D
0047  B1 FA
0049  C9 0D
004B  F0 E6
004D  D5 E0
004F  D0 EF
0051  E8
0052  D0 EE
0054  A5 DF
0056  D0 19
0058  20 2F 1E
005B  A0 01
005D  B1 FA
005F  F0 0E
0061  C9 0D
0063  F0 CE
0065  84 F9
0067  20 A0 1E
006A  A4 F9
006C  C8
006D  D0 EE
006F  F0 8F

0071  A0 02
0073  B1 FA
0075  F0 19
0077  C9 0D
0079  F0 03
007B  C8
007C  D0 F5
007E  84 DF
0080  A4 DF
0082  B1 FA
0084  F0 0A
0086  A0 00
0088  91 FA
008A  20 63 1F
008D  18
008E  90 F0
0090  20 C2 01
0093  E6 DF
0095  D0 09
0097  F0 25
0099  20 5A 1E
009C  C9 08
009E  D0 0E
00A0  38
00A1  A5 FA
00A3  E9 01
00A5  85 FA
00A7  B0 02
00A9  C6 FB
00AB  18
00AC  90 EB
00AE  C9 1B
00B0  F0 12
00B2  C9 0D
00B4  D0 05
00B6  20 2F 1E
00B9  A9 0D
00BB  C8
00BC  91 FA
00BE  20 63 1F
00C1  18
00C2  90 D5
00C4  C8
00C5  A9 3E
00C7  91 FA
00C9  20 63 1F
00CC  98
00CD  91 FA
00CF  20 63 1F
00D2  A5 FA
00D4  8D F7 17
00D7  A5 FB
00D9  8D F8 17
00DC  4C 00 00

0100  A9 00
0102  85 FA
0104  A9 02
0106  85 FB
0108  A9 7F
010A  8D 41 17
010D  A9 13
010F  8D 42 17
0112  20 41 1A
0115  46 F3
0117  05 F3
0119  85 F3
011B  8D 40 17
011E  C9 16
0120  D0 F0
0122  20 24 1A
0125  8D 40 17
0128  C9 2A
012A  D0 F2
012C  20 24 1A
012F  C5 F4
0131  F0 0C
0133  20 8C 1E
0136  20 A0 1E
0139  20 9E 1E
013C  4C 00 01
013F  91 FA
0141  20 63 1F
0144  20 24 1A
0147  AA
0148  D0 F5
014A  91 FA
014C  20 8C 1E
014F  20 1E 1E
0152  4C 00 00
0155  00
0156  00
0157  A9 00
0159  85 FA
015B  A9 02
015D  85 FB
015F  A9 27
0161  85 F5
0163  85 F4
0165  A9 BF
0167  8D 43 17
016A  A9 16
016C  20 88 01
016F  C6 F4
0171  D0 F7
0173  A9 2A
0175  20 88 01
0178  A0 00
017A  B1 FA
017C  20 63 1F

017F  48
0180  20 88 01
0183  68
0184  D0 F2
0186  F0 C4
0188  A0 07
018A  84 F2
018C  A0 02
018E  84 F3
0190  BE BE 01
0193  48
0194  20 47 17
0197  10 FB
0199  B9 BF 01
019C  8D 44 17
019F  A5 F5
01A1  49 80
01A3  8D 42 17
01A6  85 F5
01A8  CA
01A9  D0 E9
01AB  68
01AC  C6 F3
01AE  F0 05
01B0  30 07
01B2  4A
01B3  90 DB
01B5  A0 00
01B7  F0 D7
01B9  C6 F2
01BB  10 CF
01BD  60
01BE  02 C3
01C0  03
01C1  7E A9 2A
01C4  20 A0 1E
01C7  20 2F 1E
01CA  60
01CB  20 5A 1E
01CE  C9 7E
01D0  D0 0B
01D2  20 5A 1E
01D5  85 F4
01D7  20 2F 1E
01DA  4C 00 01
01DD  C9 5E
01DF  D0 0C
01E1  20 5A 1E
01E4  8D 00 02
01E7  20 9E 1E
01EA  4C 57 01
01ED  4C 1D 00
```

Dieses KIM-Programm sucht Textblöcke nach Stichworten. Die Startadresse ist 0000

Ursula und Lutz May

Mikrocomputer sind inzwischen recht preiswert geworden; für rund 200 DM bekommt man heute schon einfache Systeme, auf denen man eigene Programme entwickeln kann. Allerdings ist die Eingabe auf der meist mitgelieferten Hexadezimal-Tastatur und die Anzeige mit Siebensegment-Displays nicht sehr komfortabel. Um die im Vergleich zum Mikrocomputer selbst um ein Vielfaches höheren Kosten eines Terminals in vertretbaren Grenzen zu halten, beschreiben wir hier eine Möglichkeit, wie sich ein üblicher Fernsehempfänger zweckentfremden läßt und zusammen mit einer ASCII-Tastatur ein vollwertiges Terminal bildet.

Bildschirmterminal für Mikrocomputer

1. Teil

Ein Mikrocomputer ist ein kleines integriertes System, das die Funktion eines vollwertigen Computers ausführt. Die Mindestbausteine sind: Zentraleinheit (CPU), Speicher (ROM, RAM) sowie die Ein- und Ausgabe-Einheit. Diese Baugruppen sind durch eine Busstruktur miteinander verbunden (*Bild 1*). Eine minimale Mikrocomputer-System-Anordnung mit dem Mikroprozessor MC 6800 ist in *Bild 2* dargestellt.

Um mit dem Mikrocomputer von außen Verbindung aufnehmen zu können (zum Programmieren, zur Überwachung des Programmablaufs oder um Ergebnisse abzufragen), benötigt man entweder einen Fernschreiber („Teletype") oder ein Datensichtgerät (Bildschirmterminal) mit Tastatur (Keyboard).

Ein Bildschirmterminal ist eine Geräteeinheit, die den Inhalt eines Arbeitsspeichers auf einem Bildschirm, der zu dieser Geräteeinheit gehört, darstellt. Das hier beschriebene Bildschirmterminal besteht aus einer Steuerkarte, die einen Fernsehapparat ansteuert und den Inhalt eines Arbeitsspeichers, der sich ebenfalls auf der Steuerkarte befindet oder auf einer anderen Karte plaziert ist (FUNKSCHAU 1978, Heft 10), im ASCII-Code auf dessen Bildschirm darstellt, und zwar in 16 Zeilen zu je 32 Zeichen.

Es sind auch sehr einfache, graphische Abbildungen möglich. Daß der Arbeitsspeicher, (2 mal 512 x 8 bit für 2 Seiten) auf der Steuerkarte Platz gefunden hat, ist den neuestens auf dem Markt befindlichen hochintegrierten statischen RAMs zu verdanken. Die Steuerkarte ist aber so ausgelegt, daß der Arbeitsspeicher auch außerhalb der Karte liegen kann. Die Steckerbelegung ist pin-gleich mit der in Heft 10/78 beschriebenen RAM-Karte, so daß der Verdrahtungsaufwand klein gehalten werden kann.

Ein VHF-Modulator ist zum Einspeisen des Videosignals in eine Fernsehantennenbuchse gedacht, so daß ein handelsüblicher TV-Apparat verwendet werden kann, ohne einen Eingriff im Gerät vornehmen zu müssen.

Die Schaltung

Die gesamte Schaltung des Terminals (ohne ASCII-Tastatur) zeigt *Bild 3*.

Der Oszillator

Der Arbeitstakt für die Zeilen- und Bildsynchronimpulserzeugung sowie für die Arbeitsspeicherabfragung wird von einem aus drei Invertern bestehenden Oszillator (IS10) erzeugt. Die Versorgungsspannung erhält der Inverterbaustein von dem Regler 78L05, der die 5 V aus einer 12-V-Quelle erzeugt. Werden nicht allzu hohe Anforderungen an die Zitterfreiheit des Fernsehbildes gestellt, kann man auf den Spannungsregler verzichten und IS10 durch eine Drahtbrücke mit der auf der Karte schon vorhandenen 5-V-Versorgung verbinden. Damit der Oszillator keiner Temperaturdrift unterliegt und ein völlig ruhiges Bild erzeugt, ist ein 6,6144 MHz-Quarz vorgesehen. Möchte man diesen Aufwand nicht treiben, wird statt des Quarzes eine Drahtbrücke eingesetzt und zum C-Trimmer ein Keramikkondensator von 22 pF parallel geschaltet. Man sollte aber dann hochwertige Bauelemente verwenden, um den Temperatureinfluß klein zu halten.

Mit dem Oszillator wird gleichzeitig eine negative Spannung von −3 V erzeugt. Sie wird, ebenso wie die vom Transistor BCY 59 kommende +12-V-

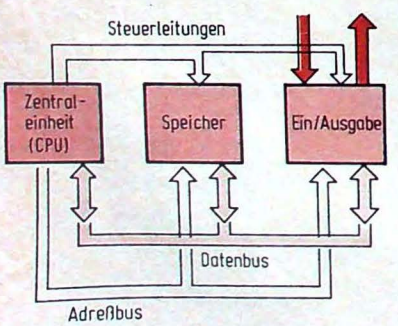


Bild 1. Prinzipaufbau eines Mikrocomputer-Systems

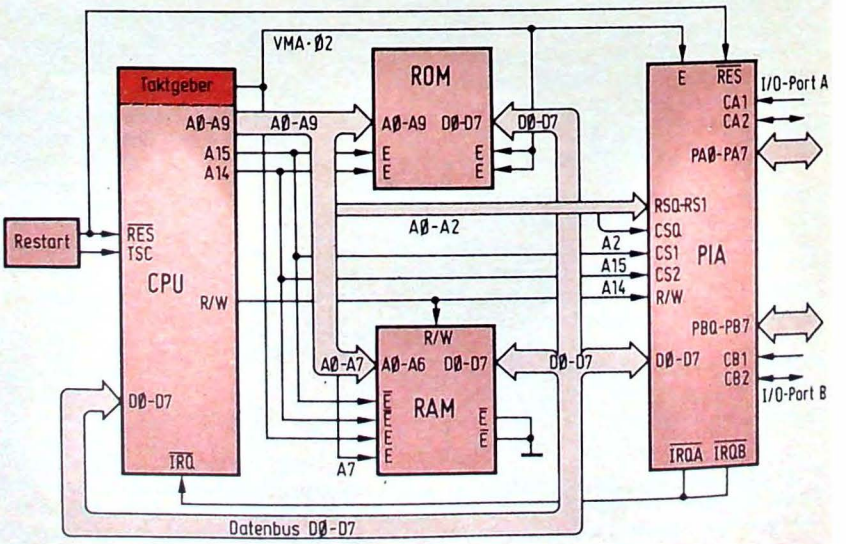


Bild 2 ▶
Minimal-Konfiguration eines 6800-Systems. Das hier beschriebene Terminal erlaubt den gleichzeitigen Zugriff von CPU und Video-Ausgabeschaltung auf das RAM; der Mikroprozessor kann so direkt auf den Bildschirm zugreifen

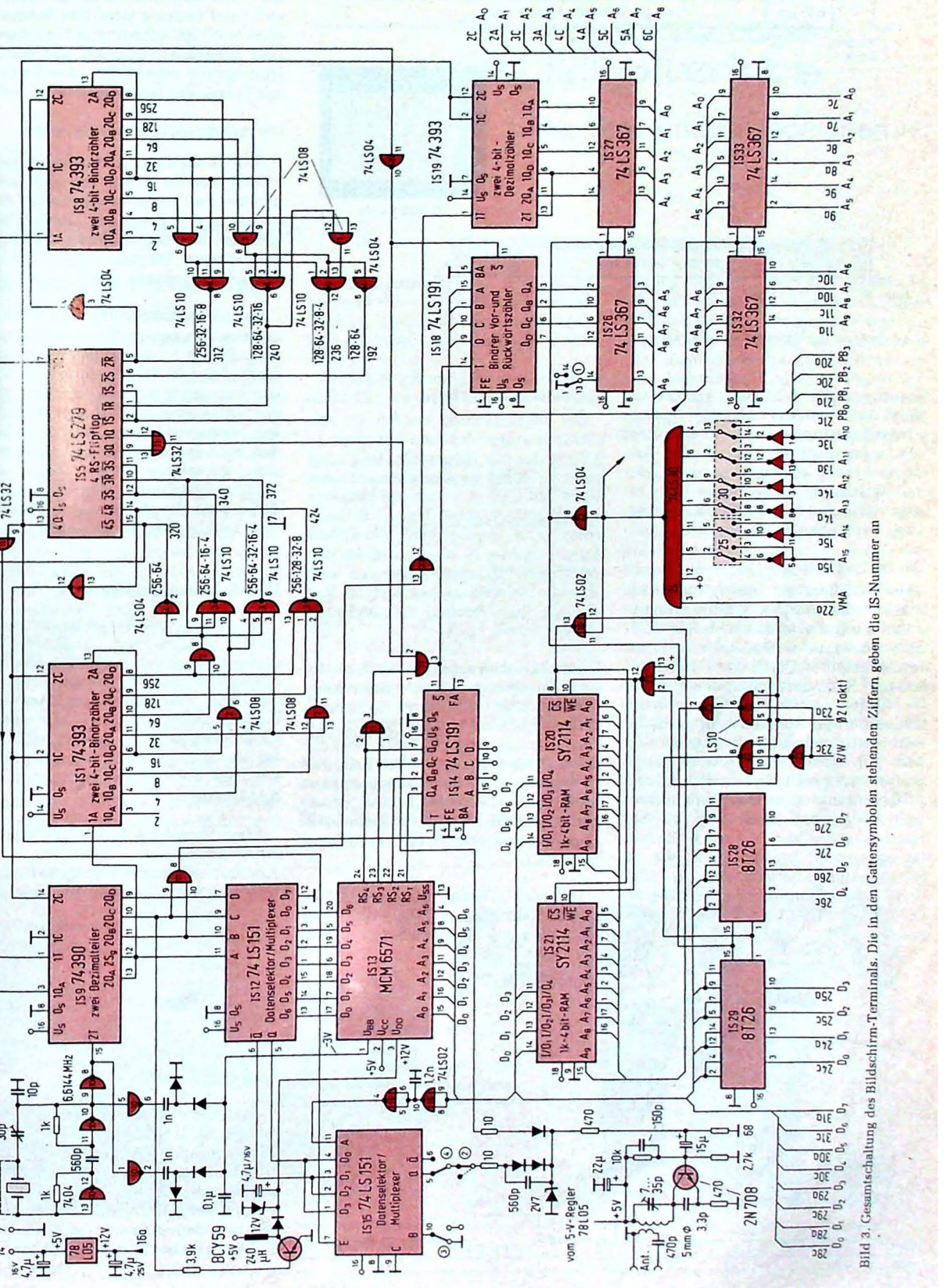


Bild 3. Gesamtschaltung des Bildschirm-Terminals. Die in den Gattersymbolen stehenden Ziffern geben die IS-Nummer an

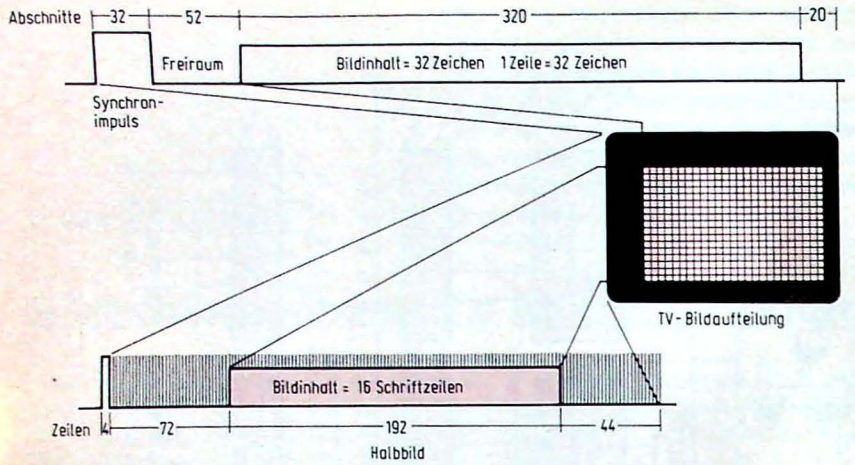


Bild 4. Struktur des Video-Ausgangssignales

Spannung, für den Zeichengenerator MCM 6571 benötigt. Alle drei Versorgungsspannungen, +5V, +12V und –3V, müssen gleichzeitig an den Zeichengenerator angelegt werden. Deshalb werden sie auch auf der Karte erzeugt, um eine fehlerhafte Netzversorgung von außen zu vermeiden.

Die Bildaufteilung

Der Oszillatortakt wird im zweifach Dezimalteiler IS 9 (SN 74390) heruntergeteilt und versorgt damit den Datenselektor-Multiplexer IS 12 sowie mit dem durch 2 geteilten Oszillatortakt den Zeilen- und Bildsynchronimpulsgenerator (IS 1 bis IS 8 und IS 11). Dort werden die Bildgröße, die Zeichenbreite und die Synchronimpulsdauer bestimmt. So wird z.B. eine Zeile in 424 Einzelabschnitte aufgeteilt, wovon 320 für den Bildinhalt und 32 für den Synchronimpuls vorgesehen sind (*Bild* 4). Ein Halbbild selbst besteht aus 312 Zeilen. Davon werden in 192 Zeilen die 16 Schriftzeilen untergebracht. Die Bild- und Synchronimpulse werden über das ODER-Gatter (IS 11, Pin 6) zum Mischer geleitet.

Der Zeichengenerator

Die Schriftzeichen werden im ASCII-Code aufgerufen und mit einer 7x 9-Punktematrix dargestellt. Insgesamt lassen sich mit dem Zeichengenerator MCM 6571 (IS 13) 128 verschiedene Schriftzeichen darstellen, z. B. auch griechische Buchstaben. Über die Adreßeingänge A_0 bis A_6 wird das darzustellende Zeichen abgerufen und über die Select-Eingänge RS1 bis RS4 die von dem Zeichen darzustellende Zeile.

Jedes Zeichen setzt sich aus 9 untereinanderstehenden Zeilen zusammen, und jede Zeile besteht aus 7 nacheinandergeschriebenen Punkten (*Bild* 5). An den Datenausgängen D_0 bis D_6 steht die Information der angewählten Zeile zur Verfügung. Mit dem Datenselektor-Multiplexer (IS 12, SN 74LS151) werden die 7 Datenpunkte der Reihe nach abgefragt und über IS 15 dem Mischer zugeführt.

Da nur 7 Datenleitungen zur Zeichenanwahl genügen, bleibt die 8. Datenleitung des Datenbus frei. Sie dient zur Hintergrunddarstellung und führt zum Datenselektor IC 15. Je nach Zustand der Leitung wird das Schriftzeichen weiß auf schwarzem Hintergrund oder umgekehrt schwarz auf weißem Hintergrund abgebildet. Man kann damit Textstellen besonders hervorheben.

Die Arbeitsspeicheradressierung

Zwei Binärzähler (IS 18 und IS 19, SN 74393) werden durch den Zeichengenerator- und Zeilensynchrontakt angesteuert und wählen aus dem Arbeitsspeicher diejenige Adresse an, die zum momentanen Darstellungsort auf dem Bildschirm gehört.

Da die Adreßdecodierung und Datenselektierung aus den RAMs eine gewisse Zeit in Anspruch nimmt, wird der Binärzähler IS 19 um einige Mikrosekunden vorher angesteuert. Dazu dient das UND-Gatter IS 17. Kurz vor Darstellungsende des letzten Zeichens löst es den nächsten Zähltakt aus. Verwendet man schnellere RAMs, kann diese Taktvoreinstellung wegfallen; allerdings sind die Preise für RAMs stark von der Zugriffszeit abhängig.

Der Arbeitsspeicher

Als Arbeitsspeicher dient die in Heft 10/78 beschriebene 1k-RAM-Karte, wobei 1/2 k genügen würden. Die Bildschirmterminalkarte ist so ausgelegt, daß man den Arbeitsspeicher ohne Hardwareänderungen auf die gleiche Karte legen kann. Es handelt sich hierbei um den gleichen Speicheraufbau wie bei der schon genannten 1-k-RAM-Karte.

Über Miniaturschalter läßt sich die Ansprechadresse einstellen. Der Adreßdecodierraum beträgt 64 k und ist für einen 16-bit-Adressenbus bestimmt.

Da durch die verwendeten Speicher-IS der Speicherbereich nicht zu verkleinern ist, sich aber mit dieser Schaltung nur 1/2 k auf dem Bildschirm darstellen läßt, kann mit dem Schalter am Adreßbuffer (IS 26, SN 74LS367) der Speicherbereich gewählt werden. So wird das 1-k-RAM auf zwei Bildschirmseiten dargestellt.

Man kann aber auch diesen verbleibenden 1/2-k-Speicherteil als normalen Programmspeicher verwenden. Dabei ist allerdings zu beachten, daß sich bei zu häufigem Rechnerzugriff kleine Störungen auf dem Bildschirm bemerkbar machen, da immer nur ein System den Arbeitsspeicher ansprechen kann.

(Schluß folgt

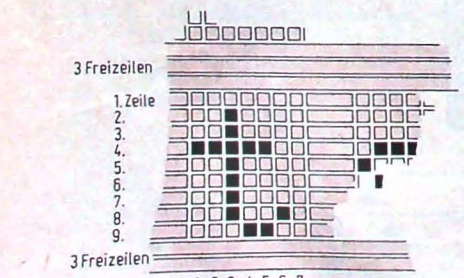
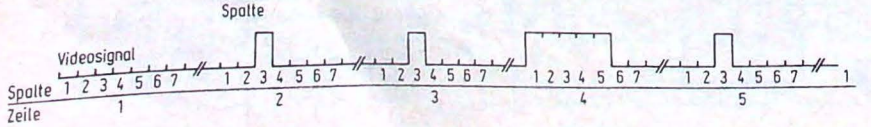


Bild 5. Prinzip der Zeichendarstellung; hier ist das Zustandekommen des kleinen Buchstabens „t" gezeigt



Hans Werner

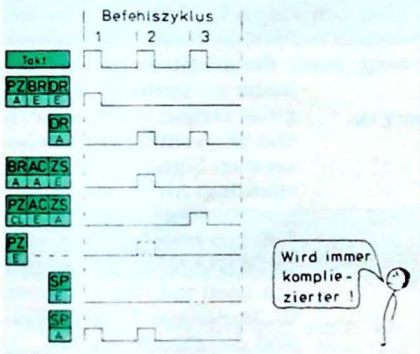
Mikrocomputer

Eine kleine Einführung

5. Teil

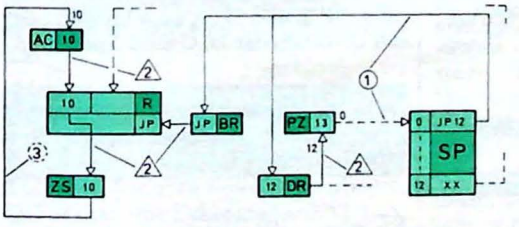
Kleine Wiederholung

Der Impuls PZ, E kommt nur mit dem Befehl JP; in allen anderen Fällen nicht!



Intervall 1

PZ adressiert Speicherzelle 0, und der Befehlsteil von JP 12 wird in das Befehlsregister, der Adreßteil in das Datenregister übernommen.



Intervall 2

Ausgabe von Befehls- und Datenregister. Der Inhalt des Datenregisters wird nicht zum Adressieren des Speichers benutzt, sondern durch den Impuls PZ, E vom Programmzähler eingelesen.

Intervall 3

Umspeichern vom ZS in den Akku und Erhöhen des Programmzählers um 1. Dies ist auch der Grund dafür, daß die Sprungadresse um 1 kleiner gewählt werden muß.

Eine Frage:

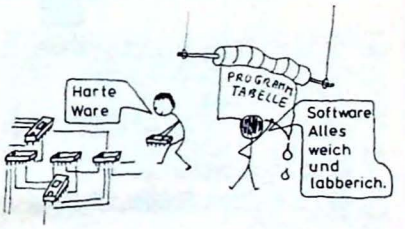
Warum kann bei unserem Modellcomputer nicht nach Adresse 0 gesprungen werden?

Lösung:

Weil dazu der negative Wert JP-1 angegeben werden müßte. Beispiel: Ein Programm beginnt mit LA 20. Es soll ein Sprung zum Anfang erfolgen. Das geht nur, wenn Adresse 0 mit einem wirkungslosen Befehl belegt wird und das Programm mit 1 beginnt.

```
0 LA 20      Das Programm beginnt
1 LA 20
2 :          mit Befehl 1.
11 Sprung    Befehl 0 hat keine Bedeutung.
```

Zwischendurch zwei Fachausdrücke:



AUSSPRACHE

Hardware: Hardwär

Software: Softwär

Die Hardware...

ist die gesamte technische Ausrüstung eines Computers, also alle Bausteine und Bauelemente.

Als Software...

werden das Programm und der Befehlsvorrat bezeichnet. Ohne Programm kann kein Computer arbeiten. Hardware und Software gehören zusammen.

Das erste Programm

Bisher haben sich alle Abschnitte mit der Hardware beschäftigt. Damit das Computermodell nicht zu abstrakt erscheint, wird zwischendurch ein einfaches Programm eingeflochten.

Problem:

Die Stellung von 2 Schaltern a und b wird durch den Inhalt von 2 Speicherzellen simuliert.

Schalter	Inhalt Speicherzelle
offen	00000000
zu	00000001

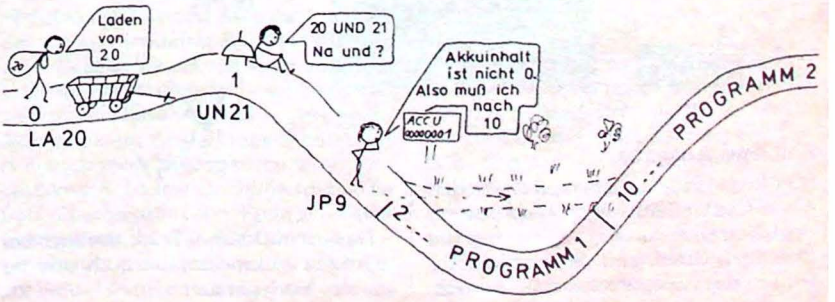
Wenn auch nur einer der Schalter geöffnet ist, soll Programmteil 1 durchlaufen werden. Sind beide Schalter geschlossen, soll Programmabschnitt 1 übersprungen und Abschnitt 2 ausgeführt werden.

Festlegungen:

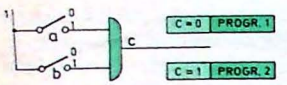
a) Schalterstellung a wird durch Zelle 20, b durch Zelle 21 dargestellt.
b) Programmteil 2 beginnt mit Adresse 10.

Lösung:

Schritt 0: Inhalt von Speicherzelle 20 in den Akku laden (Schalter a).
Schritt 1: UND-Verknüpfung von Akkuinhalt mit Inhalt Zelle 21 (a ∧ b). Das Ergebnis steht danach im Akku.



Schritt 2: Sprung nach Adresse 10, wenn der Akkuinhalt ungleich 0 ist. Nur dann sind beide Schalter geschlossen (s. Tabelle).
Schritt 3: Beginn von Programmteil 1.



a	b	c	PROGR.
0	0	0	1
0	1	0	
1	0	0	
1	1	1	2

Programm:

Kurzform	wirkliche Form
0 LA 20	0 100 10100
1 UN 21	1 010 10101
2 JP 9	2 110 01001
3 Progr 1	3 Pro gramm 1
10 Progr 2	10 Pro gramm 2

Nur wenn Inhalt AC ungleich 0, sonst geht es mit 3 weiter.

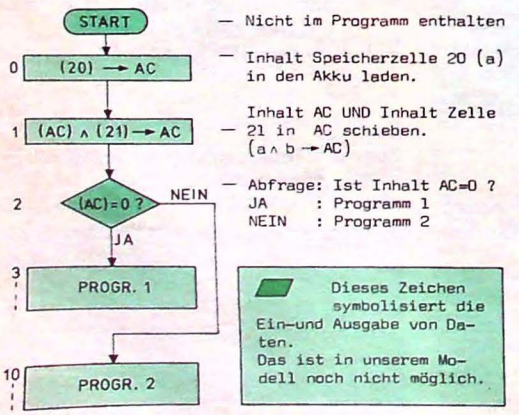
Das Flußdiagramm

Flußdiagramme vereinfachen die Darstellung von Problemen; sie können einfach in Pfeilrichtung gelesen werden.

Die Raute ist das Symbol für eine Verzweigung. An dieser Stelle sind 2 Wege möglich.

Die Beschriftung kann in normaler Sprache oder in symbolischer Schreibweise erfolgen. Für das vorstehende Beispiel wurde eine symbolische Darstellung gewählt. Klammern sollen immer Inhalt, Pfeile immer Datentransport bedeuten.

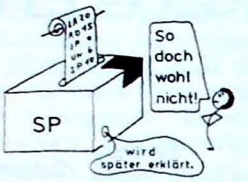
Beispiel: AC = Akku
(AC) = Inhalt von Akku



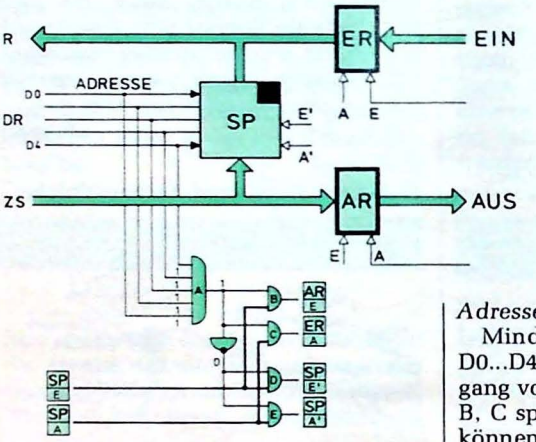
Ein- und Ausgabe

In dem Programmbeispiel wurden die Schalter stillschweigend durch Speicherzellen ersetzt. Nun möchte man auch Daten (z. B. Schalterstellungen) in das Computersystem hineingeben bzw. die Ergebnisse nach außen führen. Ohne solche Möglichkeiten ist jedes Gerät sinnlos.

Ein weiteres Problem ist: Wie kommt das Programm in den Speicher? Dieses hat nichts mit dem eigentlichen Mikrocomputer zu tun, denn das Programm



wird mit Sondereinrichtungen in den Speicher eingeschrieben. Das fertige, mit einem Mikrocomputer ausgestattete Gerät hat normalerweise die Einrichtung nicht mehr.



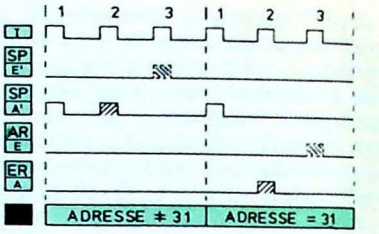
Für die Ein- und Ausgabe von Daten werden 2 Register vorgesehen. Sie vermeiden einen unkontrollierten Datenaustausch mit der „Außenwelt".

ER Eingangs-Register

AR Ausgangs-Register

Ein- und Ausgabe erfordern 2 zusätzliche Befehle, die unser Modell leider nicht besitzt. Alle $2^3 = 8$ möglichen Befehlskombinationen sind belegt.

Da hilft ein kleiner Trick, der auch bei richtigen Mikrocomputern ab und zu aus der Verlegenheit hilft.



Ein- und Ausgaberegister sind einfach parallel zum Speicher geschaltet. Damit Speicher und Register nicht gleichzeitig Daten aufnehmen bzw. ausgeben, werden sie gegeneinander verriegelt. Außerdem müssen noch die Befehle AUSGABE und EINGABE künstlich erzeugt werden. Beides geschieht durch eine Kombinationsschaltung, durch die die Steuerbefehle entweder an Speicher oder Register verteilt werden. SP, E und SP, A sind die schon bekannten Signale. Sie führen allerdings nicht mehr direkt zum Speicher.

Die Befehle LADEN und SPEICHERN funktionieren nur noch mit den Adressen 0...30. Adresse 31 ist für Ein- und Ausgabe reserviert.

Adresse 0...30

Mindestens ein Adreßeingang D0...D4 ist 0 und damit auch der Ausgang von UND A, d.h. die UND-Glieder B, C sperren und die Register ER, AR können keine Daten ausgeben bzw. aufnehmen. Dagegen gelangen die Impulse SP, E und SP, A ungehindert über die UND-Glieder D, C zum Speicher.

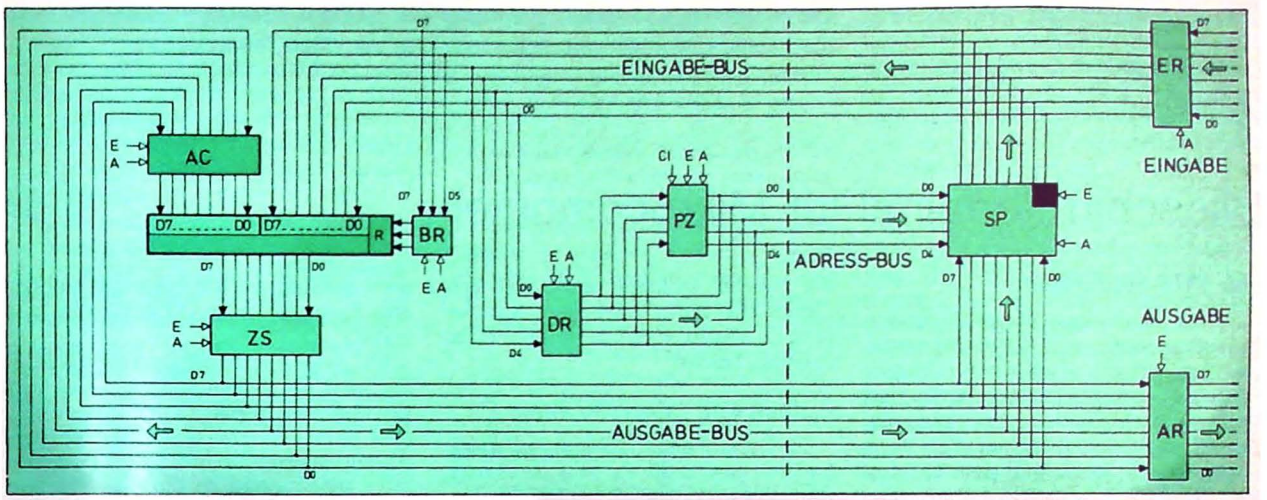
Der Ablauf der Befehle LA und SP ändert sich nicht bei Adressen ungleich 31 (≠ 31).

Eingabe 100 11111
Kurzschreibweise: LA 31

Der Eingabebefehl ist identisch mit dem Ladebefehl. Nur muß Adresse 31 angegeben werden. Da erst mit Intervall 2 die Adresse aus dem Datenregister zum Speicher gelangt, passiert der 1. SP, A-Impuls das UND E. Danach sperren die Gatter D und E, weil alle Adreßeingänge 1 werden. Der 2. SP, A-Impuls steuert das Eingaberegister, dessen Inhalt in den Akku geschoben wird.

Ausgabe 101 11111
Kurzschreibweise: SP 31

Wie beim SPEICHERN gelangt der Inhalt des Akkus auf den Ausgabebus. Der Speichereingang ist gesperrt und dafür das Ausgaberegister aufnahmebereit (SP, E = 0, AR, E = 1).



Gesamtschaltung des bisher verwendeten Mikrocomputer-Modells

(Fortsetzung folgt)

µC-Nachrichten

Auch **Sharp** brachte jetzt einen Mikrocomputer-Bausatz heraus. Zur Zeit ist er allerdings erst in Japan zu haben und kostet umgerechnet rund 250 DM. Der Computer enthält eine Hexadezimal-Anzeige und eine kleine Tastatur. Ein Monitorprogramm ist in 1 KByte ROM untergebracht, und an RAM befinden sich 256 Byte auf der Platine. Ob und wann das Produkt auch in Europa zu haben ist, läßt sich noch nicht absehen.

□

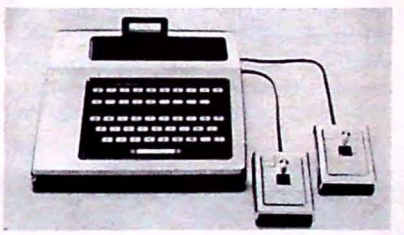
Auch in **Indien** wurde jetzt ein Mikrocomputer entwickelt, nämlich vom National Aeronautical Laboratory. Die 8-bit-CPU kann 48 Befehle verarbeiten und besitzt eine Zykluszeit von 12,5 µs; die Adressierfähigkeit ist 16 kByte.

□

Texas Instruments fertigt nun bereits ein dynamisches **64-kbit-RAM**. Es kommt mit einer 5-V-Versorgung aus und benötigt nur rund 200 mW. Ein „Refresh" ist alle 4 ms erforderlich. Die Zugriffszeiten liegen im Bereich von 100...150 ns bei minimalen Zykluszeiten von 200...250 ns. Die Fotomasken für den Chip wurden unter Anwendung der Elektronenstrahl-Lithographie gefertigt.

□

Von **Philips** wird nun ebenfalls ein Heimcomputer angeboten; das Gerät heißt G 7000 (*Bild*) und eignet sich für den Anschluß an Farbfernsehgeräte.



Der Philips-Heimcomputer G 7000 arbeitet mit ROM-Kassetten als Programmspeicher und verdiente daher eher die Bezeichnung Bildschirmspiel

Der eingebaute Mikroprozessor ist der 8048 von Intel/Signetics mit 1 KByte ROM auf dem Chip. Der Heimcomputer ist vor allem für Bildschirmspiele gedacht; passende ROM-Kassetten (ähnlich wie bei dem System von Fairchild/Saba) besitzen jeweils 2 KByte Speicherkapazität und sollen knapp 50 DM kosten. Auch der Preis des Computers selbst ist mit etwa 600 DM recht interessant.

□

Bei der Instrumentierung von 28 Tieföfen der Thyssen AG setzt Siemens den ersten **Lichtleiter-Bus** der Welt ein. Jeder der Öfen wird von einer autarken Mikrocomputer-Station gesteuert. Jeweils zwei Stationen sind so gekoppelt, daß bei einem Ausfall eine Station zur Not auch zwei Öfen steuern kann, um größeren Schaden zu vermeiden. Die Datenübertragung per Lichtleiter-Bus wurde vom Institut für Informationsverarbeitung in Technik und Biologie der Fraunhofer-Gesellschaft entworfen.

Rund **500 Programme** für die verschiedensten Mikrocomputer, darunter auch für den PET, und für die unterschiedlichsten Anwendungen zu Hause und im „Small-Business"-Bereich vertreibt die Firma Hofacker in Holzkirchen. Die Programme werden zum Großteil in Form von Magnetband-Kassetten geliefert und sind so konzipiert, daß sie sich nach dem Programmstart selbst erklären – so erscheinen z.B. bei den Spielprogrammen zunächst einmal die Spielregeln in Klartext auf dem Bildschirm.

□

Was früher als **„Heimcomputer"** angekündigt worden war – wie PET 2001 oder TRS-80 – wird jetzt still und leise als Hobbycomputer oder auch für Anwendungen im geschäftlichen Bereich („Small Business") vertrieben, nachdem sich herausgestellt hat, daß der Markt für echte Heimcomputer, etwa um der Hausfrau die Arbeit zu erleichtern, noch nicht reif ist. Hier irrte also Commodore – bei Radio Shack hat man es bereits eingesehen, wie aus der Art der Anzeige für den TRS-80 eindeutig hervorgeht (vgl. Television Digest Vol. 18, No. 35).

□

Als ersten in Europa hergestellten BASIC-Mikrocomputer bezeichnet **Corex** das Modell **Proteus III**. Es besitzt die Größe einer Reiseschreibmaschine, läßt sich an jedes Fernsehgerät anschließen und verfügt über 16 kByte RAM und 8 kByte ROM für den BASIC-Interpreter. Der Proteus III soll knapp 4000 DM kosten und arbeitet mit dem Mikroprozessor 6800.

Der zweite Teil des Beitrages beschäftigt sich mit Fragen des Aufbaus und der Inbetriebnahme sowie mit Anwendungsbeispielen der Video-Terminal-Karte

Ursula und Lutz May

Bildschirmterminal für Mikrocomputer

2. Teil

Der VHF-Modulator

Damit man ohne Eingriffe in den Fernsehapparat die Bildinformation übertragen kann, wird hier ein kleiner „Sender" verwendet. Sein Aufbau ist völlig unproblematisch. Die Frequenzeinstellung geschieht an der mit 9 Wdg. angegebenen Spule und dem C-Trimmer. Sind die Kontrasteinstellungen zu kritisch, muß man den Dämpfungswiderstand vor dem 15-µF-Koppelkondensator (68 Ω) ändern.

Damit sich Störungen von der 5-V-Spannungsversorgung nicht unangenehm bemerkbar machen, wird die Versorgung ebenso wie beim Taktoszillator vom + 5-V-Regler entnommen, worauf – bei niedrigeren Ansprüchen – verzichtet werden kann.

Über eine möglichst kurz zu haltende abgeschirmte Leitung wird die Hf-Information entnommen und zur Antennenbuchse des Empfängers geleitet. Verwendet man einen Monitor, kann der Sender entfallen und man entnimmt die Daten nach dem Mischer (470 Ω). Die Bildqualität ist dann meist deutlich besser.

Aufbau und Inbetriebnahme

Die *Tabelle* nennt die zum Aufbau erforderlichen Teile. Es ist von Vorteil, wenn beim Aufbau IS-Sockel verwendet werden. So können die einzelnen Baugruppen nacheinander in Betrieb genommen und geprüft werden. *Bild 6* zeigt die Bestückung der (fertig erhältlichen, durchkontaktierten) Platine.

Je nach Ausführung (mit Spannungsregler 7805 oder ohne) muß zur + 5-V-Versorgung eine 12-V-Spannung an Pin 16a gelegt werden. Wenn ein Frequenzzähler vorhanden ist, wird damit der Taktoszillator (IS 10) auf seine Arbeitsfrequenz von 6,6144 MHz eingestellt. Bei Verwendung eines entsprechenden Quarzes ist diese Einstellung nicht notwendig. Eine ungenaue Frequenzeinstellung hat ein nicht stehendes Fernseh- oder Monitorbild zur Folge.

Mit dem C-Trimmer im VHF-Oszillator wird die Arbeitsfrequenz auf den gewünschten Fernsehkanal eingestellt. (Mit dem Senderausgang nur an den Antennenanschluß eines Fernsehapparates gehen, *nicht* an die Hausantenne anschließen!) Sind alle notwendigen Einstellungen vorgenommen, muß auf dem Bildschirm eine einheitliche, unverrauschte Fläche sichtbar sein. Dann ist zu prüfen, ob die drei Versorgungsspannungen für den Zeichengenerator vorhanden sind (MCM 6571, IS 13): Pin 1 ca. –3 V, Pin 2 genau +5 V und Pin 3 etwa + 12 V.

Im ausgeschalteten Zustand schließt man nun den Zeichengenerator an (24poligen IS-Sockel verwenden). Nach dem Einschalten bildet der Bildschirm, je nach Zustand der Datenleitungen, 512 Zeichen in 16 Zeilen ab. Mit den

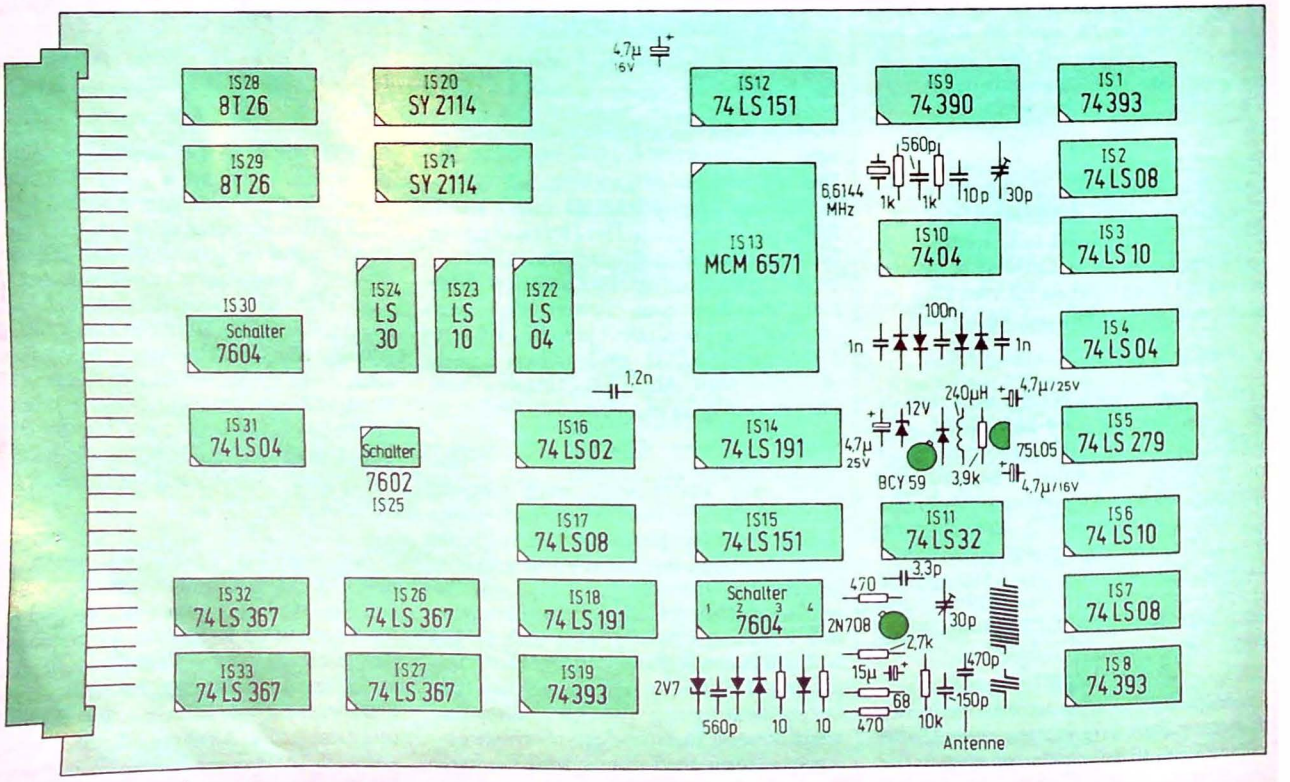


Bild 6. Bestückungsplan

Schaltern 3 und 4 läßt sich die Darstellungsart (schwarz-weiß oder weißschwarz) wählen. Schalter 2 dient zum Abtrennen des Videosignals. Der Sender wird nur noch mit den Synchronimpulsen versorgt. Über diesen Schalter läßt sich eine andere Bildinformation darstellen.

Der zwischen Pin 5, 6 und 10 (IS 16) zur Masse gelegte Kondensator dient zur Symmetrierung der Hintergrunddarstellung eines Zeichens. Da die Auswertung der Datensignale im Zeichengenerator länger dauert als das Hintergrundsignal D 7 über IS 16, muß dieses Signal verzögert werden. Eventuell ist der 1,2-nF-Kondensator im Wert zu verkleinern.

Soll der Arbeitsspeicher auf der gleichen Karte verwendet werden, müssen noch zusätzlich folgende IS eingesetzt werden:

IS 20,21	SY 2114	1k x 4 bit RAM
IS 22,31	74LS04	Inverter
IS 23	74LS10	3fach NAND-Gatter
IS 24	74LS30	8fach NAND-Gatter
S 25,30	Miniatur-Schalter	7602, 7604
IS 28,29	8T26	Datenbuffer
IS 32,33	74LS367	Adreßbuffer

Die Arbeitsweise dieses Arbeitsspeichers wurde in groben Zügen schon in Heft 10/78 FUNKSCHAU beschrieben.

Die Speicher-IS werden im Betriebszustand sehr heiß, was auf ihre hohe Packungsdichte zurückzuführen ist. Bei der Plazierung der Bildschirmterminalkarte ist auf gute Durchlüftung zu achten.

Mit den Miniaturschaltern wählt man die Ansprech- oder Arbeitsadresse des Speichers. Der verbleibende Schalter 1 hat die Funktion, den darzustellenden Speicherbereich zu wechseln („Seite umblättern").

Tabelle: Verwendete Spezialteile

Halbleiter
Z-Dioden 2,7 V und 12 V; BCY 59; 2 N 708; 8 x 1 N 4148; 7404, 2 x 74LS04, 3 x 74LS08, 2 x 74LS10, 74LS32, 2 x 74LS151, 2 x 74LS191, 74LS279, 2 x 74LS367, 4 x 74393, MCM 6571 (Sasco)

Quarz
6,6144 MHz, Miniaturgehäuse (bei den Verfassern erhältlich)

Miniaturschalter 7604 (Werdenfels)

64poliger Leistenstecker C42234A, DIN 41612 (Siemens)

Platine
geätzt, durchkontaktiert und verzinnt (bei den Verfassern erhältlich)

Für den Arbeitsspeicher zusätzlich benötigt:
IS: 2 x 74LS04, 74LS10, 74LS30, 2 x 74367, 2 x 8T26 (Sasco)
Miniaturschalter 7602 und 7604 (Werdenfels)
RAM-IS: 2 x SY 2114 (Astronic)

Als mögliche Fehlerquellen bei der Inbetriebnahme kommen in Frage:

a) Unsauberes Schwingen des Taktoszillators IS 10: 560 pF oder 10 pF ändern
b) Kein Fernsehbild: Transistor 2N708 wechseln, Versorgungsspannung prüfen
c) Verwackeltes Bild (Brumm): Netzteil besser sieben oder Quarz (6,6144 MHz) und/oder den auf der Platine vorgesehenen Spannungsregler 78L05 verwenden.
d) keine Zeichendarstellung: Versorgung des Zeichengenerators prüfen, Schalterstellung 2 überprüfen.

Anwendungsbeispiele

Die Video-Terminal-Karte läßt sich recht universell anwenden:

- Bildschirmsteuerkarte (Bildschirmterminal ohne Arbeitsspeicher)
- Bildschirmterminal (mit Arbeitsspeicher)
- Einfache 1k x 8-bit-Speicherkarte
- 1k x 8-bit-Speicherkarte, die von zwei verschiedenen Systemen angesteuert werden kann (vgl. Heft 10/78)
- Daten- und Adreßbufferkarte

Da die Daten- und Adreßbusleitungen doppelt zur Steckerleiste führen (gebuffert und ungebuffert), läßt sich die Karte – je nach Bestückung – sehr universell verwenden. Die ersten beiden Anwendungsbeispiele wurden im vorausgehenden Text ausführlich beschrieben. Da der Arbeitsspeicher einen getrennten Aufbau hat, kann auch frei über ihn verfügt werden.

Steuert der Rechner den Speicher nicht an, kann ein zweites Rechnersystem den Arbeitsspeicher ansteuern. Dazu können die beiden Buffer-IS 26/27 SN 74LS367 verwendet werden. Über zwei Dual-in-Line Stecker (16- und 14polig) erhält man den Datenzugriff zu den in diesem Fall nicht zu bestückenden IS-Sockeln 18 und 19. Spricht das erste Rechnersystem den Speicher an, wird die Verbindung zum zweiten automatisch unterbrochen.

In einem größeren Rechnersystem muß wegen der größeren Busbelastung der Daten- und Adreßbus mehrmals gebuffert werden. Verwendet man nur die Bufferbausteine IS 28/29 und IS 32/33 sowie die Steuergatter IS 22...24, kann die fast leere Bildschirmkarte als einfacher Busverstärker verwendet werden.

Unbestückte sowie bestückte und geprüfte Platinen sind von den Verfassern zu beziehen (Schlosserbreite 6, 8018 Grafing).

Was ist ein Monitor-Programm?

Es gibt im Prinzip zwei Möglichkeiten, Informationen oder Programmbefehle in den Speicher eines Mikrocomputer-Systems per Hand einzuschreiben: Entweder bringt man alle Leitungen des Mikroprozessors zum Speicher in den hochohmigen Tri-State-Zustand und gibt Adressen nebst zugehörigen Daten direkt per Hardware auf den Adressen- und Datenbus, oder aber man benutzt ein Programm, das über Eingabe-Ausgabe-Leitungen des Mikrocomputers ein Tastenfeld abfragt und die eingetippten Daten an die gewünschten Adressen abspeichert. Das Programm, das dieses tut, nennt man Monitor-Programm; es hat überhaupt nichts mit einem Fernseh-Monitor zu tun!

Während die beschriebene Hardware-Methode einfacher zu verstehen ist, bringt die Software-Methode per Monitor-Programm einige bedeutende Vorteile. Denn ohne großen Aufwand kann man dieses Programmier-Programm auch mit einigem Komfort ausstatten, den die Hardware-Logik nie bieten kann. Das Monitor-Programm kann z. B. bei einem Stop des Anwenderprogramms die internen CPU-Register an festgelegte Speicheradressen „retten", wo man sich dann ansehen kann, was die CPU zuletzt eigentlich getan hat, und kann beim Drücken einer GO- oder RUN-Taste diese Speicheradressen wieder in die CPU zurückbefördern, so daß das Anwenderprogramm weiterläuft, als wäre nichts geschehen. Oder das Monitorprogramm enthält gleich die notwendigen Routinen zur Aufzeichnung von Programmen auf einen Kassettenrecorder.

Und noch ein Vorteil eines Monitor-Programms: Es enthält meist einige sehr nützliche Unterprogramme, z. B. zur Anzeige von Zahlen auf einem Siebensegment-Display, zur Tasten-Decodierung oder zur Ausgabe von ASCII-Zeichen auf einem Fernschreiber. Und diese Unterprogramme lassen sich natürlich auch von Anwenderprogrammen her aufrufen – das vereinfacht die Programmierung gerade von Eingabe- und Ausgabe-Routinen schon sehr.

Das Monitor-Programm steht meist in ROM-Speichern, die auf der Mikrocomputer-Platine untergebracht sind und umfaßt, je nachdem, wie komfortabel es ist und wie genial es selbst programmiert wurde, etwa 1...2 kByte.

Karl Schedler

Der Nascom-1 mit der CPU Z 80 ist wohl zur Zeit auf dem Mikrocomputer-Sektor einer der besten Kompromisse zwischen dem, was er kann, und dem, was er kostet. Im Vergleich zu vielen anderen Kits mit hexadezimaler Tastatur und mickrigen 7-Segment-Anzeigen, die nur sehr begrenzt einsatzfähig sind, ist dieser Bausatz mit Video-Display und alphanumerischer Tastatur von Anfang an wesentlich vielseitiger verwendbar.

Nascom-1 – ein guter Kompromiß

Der Bausatz Nascom-1, ein Produkt aus dem United Kingdom – Großbritannien, ursprünglich von der Firma NAS in Trier (jetzt auch in München) angeboten, ist mittlerweile von mehreren Vertriebsfirmen erhältlich. Betriebsbereit ist er allerdings erst mit einem zusätzlichen Fernseher. Dazu kann man aber ohne Umbauten seinen Heimfernseher verwenden, indem man den Nascom einfach an den Antenneneingang anschließt.

Abgesehen von einigen billigen Kompromissen macht der Bausatz einen soliden Eindruck. Sämtliche IS sitzen auf Sockeln. Die Steckverbindungen zwischen Platine und Umwelt sind über DIL-Stecker realisiert. Es wäre aber dringend anzuraten, die Kabel gegen jede mechanische Belastung abzusichern, da sich besagte Stecker bereits bei scharfem Hinsehen verbiegen. Es gibt durchaus Bausätze, die auf Anhieb funktioniert haben, es gibt aber auch welche, die große Schwierigkeiten bereiten, z. B. wegen defekter IS.

Hardware

Bild 1 zeigt die Struktur des Systems. Die gesamte Schaltung befindet sich auf einer doppelseitigen, durchkontaktierten Platine (ca. 300 mm x 210 mm); mit Ausnahme der Tastatur natürlich, die übrigens schon fertig verdrahtet und bestückt geliefert wird. Die Tastatur besteht, außer der Reset-Taste, die noch aufgelötet werden muß, aus kontaktlosen Tasten, 46 an der Zahl in „Standard-QWERTY-Anordnung" (nicht so umfangreich wie bei einem richtigen Datensichtgerät, aber immerhin...).

Die Decodierung der Tasten erfolgt über einen CMOS-Zähler, der vom Monitor-Unterprogramm getaktet die Matrix-Spalten nacheinander durchschaltet. Die Zeilen der Matrix werden über Flipflops und Tri-State-Buffer auf den Datenbus gelegt und abgefragt. Das Unterprogramm kehrt dann mit dem richtigen ASCII-Zeichen im Akku zurück.

Sehr raffiniert ist auch die Bildschirmanzeige mit dem Video-RAM aus acht statischen RAM-Chips vom Typ 2102. Letztere werden per Hardware durch die Ausgänge einer Teilerkette am Adreßbus zyklisch abgefragt, und die Daten werden dem Zeichengenerator zugeführt. Dieser interpretiert den Speicherinhalt als ASCII-Zeichen und zaubert das entsprechende Symbol als 7 x 9-Matrix auf den Schirm. Eine Adresse im Video-RAM entspricht einer bestimmten Stelle auf dem 16 Zeilen à 48 Zeichen fassenden Bildschirm.

Für die CPU, die von der ganzen Bildaufbereitung entlastet ist, ist das Video-RAM wie ein normaler Schreib-Lese-Speicher anzusprechen, indem die Adreßleitungen einfach über einen Daten-Selektor umgeschaltet werden. Man kann also an jede beliebige Stelle im Bild jedes beliebige Zeichen setzten oder abfragen, wobei der Zeichengenerator ein größeres Repertoire aufweist als von der Tastatur her erzeugt werden kann, z. B. Kleinschreibung.

Das auf einen UHF-Oszillator modulierte Video-Signal kann auch direkt entnommen werden. Wer einmal mit direkt erzeugten Zeichen gearbeitet hat, dem kommen die durch geringe Bandbreite und Rauschen entstellten Zeichen über den Antenneneingang wie Hieroglyphen vor! Aber Vorsicht beim Einrichten eines Video-Eingangs bei alten Allstrom-Fernsehgeräten. Man sollte die Masseverhältnisse beachten und nur über einen Trenntrafo arbeiten. Es ist nämlich meist die Netzsicherung, die von der Halbleiterschaltung geschützt wird, und nicht umgekehrt!

Ein preiswertes Portable ist die beste Lösung.

Für die serielle Kommunikation mit der Außenwelt ist der UART 6402 zuständig, der vom Programm als normaler Ein/Ausgabe-Port anzusprechen ist. Eine normgerechte RS-232-Schnittstelle ist vorhanden. Mit am UART angeschlossen ist ein Cassetten-Interface, das allerdings zu wünschen übrig läßt: Abgesehen davon, daß es keiner Norm entspricht, ist die Störsicherheit trotz einer aufwendig erscheinenden Decoder-Schaltung nicht gerade überwältigend; es werden einfach aus der Videoteilerkette 1,9 kHz ausgekoppelt und über ein Gatter vom UART getaktet.

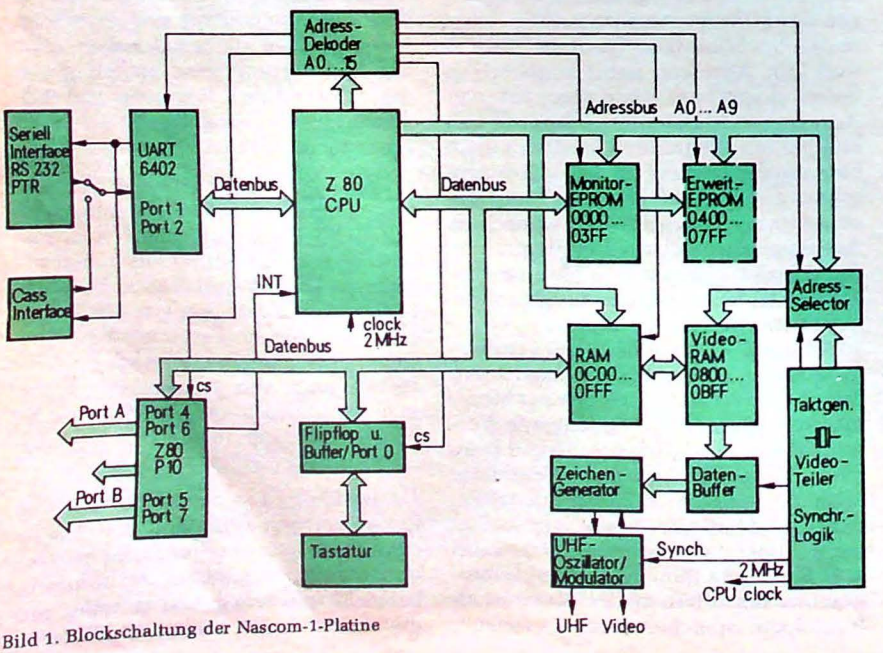


Bild 1. Blockschaltung der Nascom-1-Platine

Der Takt für den seriellen Baustein kann entweder mit 3,9 kHz (auch aus der Videoteilerkette) entsprechend 240 Baud, mit einem freischwingenden Generator (1760 Hz entsprechend 110 Baud) oder extern beschaltet werden. Es ist auch eine LED vorhanden, die das Einschalten des Recorders signalisieren soll und jederzeit vom Anwenderprogramm angesprochen werden kann. Zum automatischen Schalten kann leicht noch ein geeignetes Relais (ca. 150 Ω, 3 V) statt des Vorwiderstands angeschlossen werden. Auf dem gleichen Port sind übrigens noch zwei Ein- und Ausgänge zur freien Verfügung vorhanden.

Neben einem seriellen Interface darf natürlich eine parallele Schnittstelle nicht fehlen. Dazu wird der Standard-Z-80-PIO-Baustein verwendet. Man hat dann zwei 8-bit-Parallel-Ports, auch mit „Handshaking", zum freien Gebrauch.

Im Zusammenhang mit der Hardware sei auch noch die Single-Step-Logik erwähnt. Durch einen Ausgang am Port 0 vom Monitor gesetzt, erzeugt diese nach jedem Befehl (dazu wird der M1-Ausgang der CPU ausgewertet) einen Interrupt mit einem Sprung in das Monitorunterprogramm.

Schließlich sei noch gesagt, daß alle Daten- und Adreßbusleitungen sowie alle nötigen Steueranschlüsse (z. B. Refresh für dynamische Speicher) über Platinenstecker herausgeführt sind. Die Busleitungen müssen aber auf alle Fälle bei Anschluß weiterer Bausteine gepuffert werden. *Bild 2* zeigt die Nascom-1-Konfiguration.

Software

Das Monitor-(Betriebs-)Programm des Nascom-1 sitzt in einem 1k-EPROM (Adressen 0000...) und ist sehr komfortabel, vor allem auch flexibel. Alle wichtigen Routinen sind als Unterprogramme ausgeführt (z. B. Zeichen auf Bildschirm setzen, Tastatur abfragen und codieren, Anzeige von Speicherinhalt in Hex-Darstellung usw.) die jederzeit vom Anwenderprogramm verwendet werden können. Eine sehr nützliche Besonderheit des Monitors ist, daß nach einem Reset alle Sprungadressen der wichtigsten Unterprogramme und Tabellen an den Anfang des Benutzer-RAM gelegt werden. Das Monitorprogramm springt dann nicht direkt, sondern über das RAM ins Unterprogramm. Der Anwender hat dann alle Freiheiten, eigene Programme anspringen zu lassen, einzufügen oder Kommando- bzw. Tastatur-Codiertabellen abzuändern.

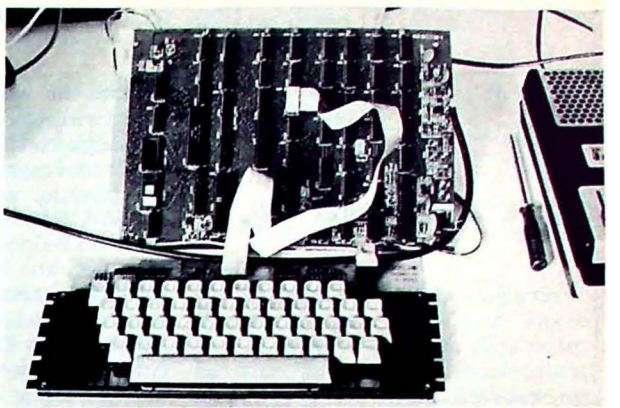


Bild 2. Der Mikrocomputer Nascom-1 besteht aus einem etwas vereinfachten Tastenfeld und der eigentlichen Computer-Platine (Foto: Fe.)

Um die umfangreichen Möglichkeiten des Betriebsprogramms ausschöpfen zu können, muß man sich allerdings schon die Mühe machen das „Listing" näher zu betrachten. Im Begleitmaterial sind leider bei weitem nicht alle Unterprogramme und Möglichkeiten aufgeführt.

Überhaupt ist das Begleitmaterial, gemessen an den Handbüchern eines KIM oder 8085-Bausatzes, wohl etwas dürftig. Obwohl man es für nötig gehalten hatte, in den „Construction notes" breit zu erklären, daß Widerstände im Gegensatz zu Dioden und Tantal-Kondensatoren durchaus auch verkehrt gepolt werden dürfen, läßt man den Anfänger dann über weit Schwierigeres im Unklaren.

Zu guter Letzt sei noch erwähnt, daß auf dem Nascom-1 als Z-80-System selbstverständlich 8080-Software verwendet werden kann. Mit fertiger Z-80- oder gar Nascom-Software sieht es bislang düster aus. Von der Firma NAS z. B. ist ein Tiny Basic und ein Assembler-Programm auf EPROM versprochen, aber bisher ist davon noch nichts zu haben. Sicher schadet es niemandem, eigenes Gehirnschmalz zu investieren; doch wer sich schon öfter mit dem Programmieren herumgeärgert hat, weiß, wie zeitraubend oft einfache Routinen sind.

Eine gute „Hintergrund-Software", die Zeit für die eigentlichen Probleme übrig läßt, wäre wünschenswert.

Wolfgang Lelle

Umwandlung von Dezimal- in Hexadezimalzahlen mit dem TI-59

Gerade in der Mikrocomputer-Technik ist es häufig erforderlich, Zahlen vom Dezimal- in das bei den meisten Mikroprozessoren übliche Hexadezimal-System umzuwandeln. Für diesen Zweck wurde ein Programm für den Taschenrechner TI-59 und den Drucker PC-100 A entwickelt, das alle Hexadezimalziffern (0...F) ausgeben kann und daher den meisten ähnlichen Programmen überlegen sein dürfte.

Der Vorteil des Programms liegt in seiner übersichtlichen Anwendbarkeit: Nach Einlesen der Magnetkarte kann ohne jede weitere Operation die zu codierende Dezimalzahl eingegeben werden. Nach Drücken der Taste A erfogt der Ausdruck: „Dezimal..." und darunter „Hex/Dez....", wobei im Hinblick auf die Anwendung einer vorhandenen Hexadezimaleingabe die Zahlen 10, 11,... bereits als A, B,... ausgedruckt werden (*Bild 1*).

Der Programmablauf ist denkbar einfach: die eingegebene Dezimalzahl wird zuerst mit der Zahl 16, dann, wenn notwendig, mit 16^2, 16^3 usw. verglichen, bis sie die entsprechende 16er-Potenz überschreitet, wobei die fortlaufende Erhöhung der 16er-Potenzen über einen bedingten Sprungbefehl in einer Programmschleife erzeugt wird und zur Kennzeichnung des Stellenwertes bei jedem Durchlaufen der Schleife der Inhalt eines Speichers (hier Speicher 04)

```
          DEZIMAL
      15.
          HEX/DEZ
        F

          DEZIMAL
  1277586.
          HEX/DEZ
        1.
        3.
        7.
        E
        9.
        2.

          DEZIMAL
2539983159.
          HEX/DEZ
        9.
        7.
        6.
        6.
        1.
        1.
        3.
        7.

          DEZIMAL
   999999.
          HEX/DEZ
        F
        4.
        2.
        3.
        F
```

Bild 1. Typischer Ausdruck bei einer Zahlenumwandlung mit dem vorliegenden Programm

um 1 erhöht wird. Ist der Stellenwert festgestellt, erfolgt die Umcodierung nach dem Euklidschen Algorithmus: die Dezimalzahl wird durch die höchste in ihr enthaltene 16er-Potenz dividiert, der Quotient mit der Operation INT ganzzahlig gemacht (nicht gerundet!) und diese ganze Zahl ausgewiesen. Durch Multiplikation dieser ganzen Zahl mit dem vorhergehenden Divisor wird durch Subtraktion dieses Produktes von der ursprünglichen Zahl der Rest gebildet und mit diesem entsprechend verfahren, wobei die 16er-Potenz, durch welche dividiert wird, vorher durch Division durch 16 um einen Stellenwert verringert wird.

Ebenso wie beim „Aufwärtszählen" wird auch jetzt wieder der Speicherinhalt des Speichers 04 beim Durchlaufen jeder Schleife um 1 verringert, so daß durch Abfragen des Inhaltes von Speicher 04 der Vorgang an der richtigen Stelle beendet wird (dem interessierten Leser, der dieses Programm optimieren möchte, sei hierzu die Anwendung der Operation DSZ empfohlen, die speziell für solche Anwendungen vorgesehen ist).

Bevor die entsprechende Zahl gedruckt wird, erfolgt eine Prüfung, ob sie größer/gleich 10 ist. Trifft dies zu, veranlaßt das Programm über die indirekte Adressierung den Aufruf eines der Speicher 10...15, in denen die Buchstaben A bis F alphanumerisch codiert gespeichert sind. Dies sei an einem Beispiel näher erläutert: die Hexadezimalstelle 12 (also C) soll gedruckt werden. Nachdem das Programm die Zahl 12 ermittelt hat, folgt die Prüfung $x \geqq 10$, die hier mit ja beantwortet wird. Dadurch folgt der Befehl STO xy und dann RCL IND xy (im Druck RC* xy), wodurch nicht der Inhalt von Speicher xy, sondern der Inhalt des Speichers aufgerufen wird, dessen Adresse als momentaner Inhalt in Speicher xy steht, in diesem Falle also Speicher 12. Speicher 12 enthält die Information 15, die durch die nachfolgenden Operationen OP 01...05 als Buchstabe C interpretiert und an richtiger Stelle gedruckt wird.

Um das Programm und Speicher auf nur eine Magnetkarte einlesen zu können (Magnetkartenseiten 3 und 4), wurde bei Speicherbereichsverteilung 719/29 (3 OP 17) mit Programmadresse 480 begonnen (siehe Betriebsanleitung des Rechners), was für die Anwendung unwesentlich ist, da der Programmstart über eine Labelmarkierung erfolgt.

Die alphanumerischen Kommentare sind in den Speichern 5 und 6 (Dezimal), in 7 und 8 (Hex/Dez.), A bis F in 10 bis 15 und die Vergleichszahl 10 in 17 gespeichert.

Anwendungen des Programms:

1. Einstellen der Speicherbereichsverteilung (3 OP 17)
2. Einlesen der Magnetkarte (Seiten 3 + 4)
3. Eingabe löschen (CLR)
4. Dezimalzahl eingeben
5. Taste (Label) A drücken

Es erfolgt der Ausdruck der eingegebenen Dezimalzahl und (von oben nach unten) der entsprechenden Hexadezimalzahl. Das Ende der Umcodierung ist durch dreimaligen Papiervorschub (ADV) erkennbar.

```
480  76 LBL    514  42 STO    548  03  03    582  02  02
481  11  A     515  04  04    549  55  ÷     583  95  =
482  42 STO    516  01  1     550  43 RCL    584  22 INV
483  01  01    517  06  6     551  02  02    585  44 SUM
484  43 RCL    518  32 X:T    552  95  =     586  03  03
485  05  05    519  43 RCL    553  59 INT    587  43 RCL
486  69 OP     520  01  01    554  42 STO    588  02  02
487  03  03    521  22 INV    555  16  16    589  55  ÷
488  43 RCL    522  77  GE    556  43 RCL    590  01  1
489  06  06    523  05  05    557  17  17    591  06  6
490  69 OP     524  39  39    558  32 X:T    592  95  =
491  04  04    525  32 X:T    559  43 RCL    593  42 STO
492  69 OP     526  65  ×     560  16  16    594  02  02
493  05  05    527  01  1     561  77  GE    595  01  1
494  25 CLR    528  06  6     562  05  05    596  22 INV
495  69 OP     529  95  =     563  69  69    597  44 SUM
496  00  00    530  32 X:T    564  99 PRT    598  04  04
497  43 RCL    531  01  1     565  61 GTO    599  01  1
498  01  01    532  44 SUM    566  05  05    600  32 X:T
499  99 PRT    533  04  04    567  78  78    601  43 RCL
500  43 RCL    534  43 RCL    568  68 NOP    602  04  04
501  07  07    535  01  01    569  73 RC*    603  67 EQ
502  69 OP     536  61 GTO    570  16  16    604  06  06
503  03  03    537  05  05    571  69 OP     605  11  11
504  43 RCL    538  21  21    572  03  03    606  43 RCL
505  08  08    539  32 X:T    573  69 OP     607  03  03
506  69 OP     540  55  ÷     574  05  05    608  61 GTO
507  04  04    541  01  1     575  25 CLR    609  05  05
508  69 OP     542  06  6     576  69 OP     610  49  49
509  05  05    543  95  =     577  00  00    611  98 ADV
510  25 CLR    544  42 STO    578  43 RCL    612  98 ADV
511  69 OP     545  02  02    579  16  16    613  98 ADV
512  00  00    546  32 X:T    580  65  ×     614  91 R/S
513  02  2     547  42 STO    581  43 RCL
```

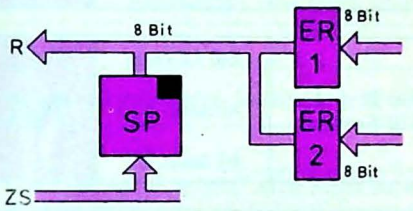
Bild 2. Das Programm zum Ausdruck hexadezimaler Zahlen auf dem Drucker PC-100. Zusätzlich müssen noch einige Datenspeicher mit bestimmten Werten geladen werden (siehe Text)

Hans Werner

Mikrocomputer

Eine kleine Einführung

6. Teil



Für einen speziellen Anwendungsfall werden 2 Eingaberegister benötigt. Ist das möglich?

☐ a) Ja, wenn eine weitere Adreßkombination geopfert wird.

☐ b) Nein, dazu ist der Modellcomputer nicht geeignet.

bei Lösung

a) nach N

b) nach I

A

Ihre Antwort ist etwas befremdlich. Zurück zur Aufgabe.

nach O

B

Sehr gut, die Lösung ist richtig.
Sie haben erkannt, daß der Zelleninhalt 000 01000 zuerst als Zahl 8 und dann als Befehl AD 8 gedeutet wird. Das gilt sinngemäß auch für 000 01001.
Wenn Sie eine genauere Erklärung wünschen, springen Sie bitte nach Punkt S.

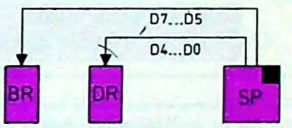
nach D

C

Was steht nach dem

Holen eines Befehls im BR .....

im DR .....



nach U

D

Das ist zwar richtig, aber in erster Linie adressiert der Programmzähler fortlaufend die Speicherplätze 0, 1, 2...
Dadurch wird ein Befehl nach dem anderen ausgeführt.
Es trifft also b und c, d.h. Lösung d zu.

nach K

E

Ihre Antwort ist falsch. Versuchen Sie noch einmal Ihr Glück.

nach K

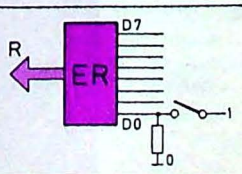
F

So ist es richtig.

Software: Programme, Befehlsvorrat des Rechenwerkes.
Hardware: Bauelemente, Leiterplatten, alle Anschlüsse usw.

nach H

G



5 LA 31
6 JP 4
7 LA 21
8 AD 22

Der Inhalt von Zelle 21 soll zum Inhalt von Zelle 22 addiert werden. In welcher Stellung muß der Schalter stehen, wenn D1...D7 fest an 0 liegen?

☐ a) geschlossen

☐ b) geöffnet

bei Lösung

a) nach W

b) nach Y

H

Zum Glück ist es doch möglich. Lesen Sie bitte die Lösung nach.

I

Sie haben richtig erkannt, daß beides zutrifft.

nach K

J

Bei welchen Befehlen bleibt der Inhalt des Akkus erhalten?

☐ a) SP
☐ b) JP
☐ c) bei beiden
☐ d) bei keinem

a) nach P
b) nach T
c) nach Q
d) nach F

K

Nach Ablauf des Befehlszyklus steht im Programmzähler die Adresse des NÄCHSTEN BEFEHLS.

nach M

L

Mit dem Programm sollen die Zahlen 8 (Zelle 2) und 9 (Zelle 3) addiert werden. Geht das?

100 = LA 000 = AD 111 = ST

0 100 00010
1 000 00011
2 000 01000
3 000 01001
4 111 00000

☐ a) Ja
☐ b) Nein

bei Lösung
a) nach S
b) nach C

M

Die von Ihnen ausgesuchte Lösung ist richtig. Mit einer weiteren Adreßkombination kann ER 2 gesteuert werden. Es ergeben sich 3 Fälle:

1. Adresse 0...29: ER 1 und ER 2 verriegelt
2. Adresse 30: ER 1 und SP verriegelt
3. Adresse 31: ER 2 und SP verriegelt

Adresse 30 wurde frei gewählt.

nach O

N

Was hat der Programmzähler für eine Aufgabe?

☐ a) Er steuert alle Register.
☐ b) Er ermöglicht den Sprungbefehl.
☐ c) Er adressiert den Speicher, so daß fortlaufend ein Befehl nach dem anderen geholt wird.
☐ d) Lösung b und c ist richtig.

a) nach B
b) nach E
c) nach X
d) nach J

O

Nicht schlecht, nur ist die Antwort nicht vollständig. Auch beim Sprungbefehl wird der Akkuinhalt nicht verändert.

nach R

P

Fein, die Antwort ist richtig. Das gilt natürlich für unseren Modellcomputer und nicht unbedingt für andere Systeme.

nach R

Q

Nach Ablauf des Befehlszyklus steht im Programmzähler die Adresse des ...

nach L

R

Leider falsch. Die Zahlen 8 und 9 stehen vor der STOP-Anweisung. Sie werden zuerst als Zahl und dann als Befehl gedeutet.

Programm:
0 LA 2
1 AD 3
2 Zahl 8 bzw. AD 8
3 Zahl 9 BZW. AD 9
4 ST

Programmablauf:
Lade Inhalt Zelle 2
Addiere Zelle 3
Addiere Zelle 8
Addiere Zelle 9
Stop

Der Inhalt der Zellen 8 und 9 ist nicht bekannt, so daß 8 + 9 + ? + ? gerechnet wird.

Richtiges Programm:
0 LA 20
1 AD 21
2 ST
20 8
21 9

Die Zahlen stehen außerhalb des Programms und können nicht als Befehl gewertet werden.
In ganz besonderen Fällen sind solche Doppeldeutigkeiten erwünscht, allerdings nicht bei unserer Aufgabe.

nach D

S

Fast richtig. Auch beim Befehl SPEICHERN wird der Akkuinhalt nicht verändert.	nach R	T

Im BR steht der Operationsteil, z. B. UN Im DR steht der Adreßteil, z. B. 8	01001000 UN 8	nach V	U

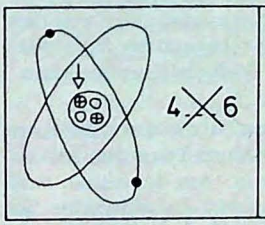
Software: Programme, Leiterplatten ohne Bauteile, flexible Anschlüsse. Hardware: Bauelemente, Befehlsvorrat des Rechenwerkes, alle Anschlüsse, Leitungen und Verbindungen.	Bitte die falschen Angaben durchstreichen.	nach G	V

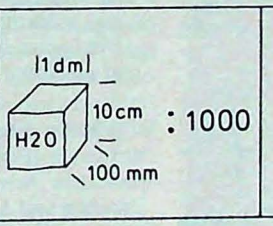
Das ist nicht richtig. Lesen Sie bitte die Lösung.	nach Y	W

Die Lösung ist richtig. Aber eines haben Sie übersehen: Durch den Programmzähler wird auch der Sprungbefehl ermöglicht. Beispiel: Bei dem Befehl JP 15 wird die Adresse 15 in den PZ geladen und das Programm mit Adresse 16 fortgesetzt. Richtige Mikrocomputer holen natürlich nicht den Befehl 16, sondern Befehl 15. Die Erhöhung um 1 ist durch die sehr einfache Steuerung des Modellcomputers bedingt.	nach K	X

Richtig! Wenn der Schalter geschlossen ist, wird eine 1 in den Akku geladen und der Sprungbefehl ausgeführt. Das Programm durchläuft dann ständig die Befehle 5, 6. Mit dieser Aufgabe ist der Test beendet. War es schlimm?	PAUSE	Y

Tante Olgas Rätselecke

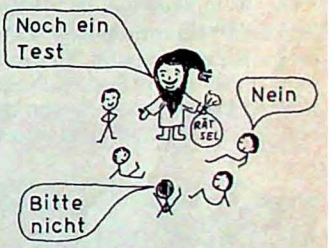




Bilderrätsel

Es ergibt sich ein Fachausdruck mit 16 Buchstaben.

Lösung:
Siehe Test 3



(Fortsetzung folgt)

µC-Nachrichten

Der erste **Ein-Chip-Codec** wurde von Intel entwickelt. Er ist in zwei Versionen zu haben, nämlich als 2910 für das amerikanische und 2911 für das europäische Telefonnetz. Die IS werden in NMOS-Technik gefertigt; sie enthalten sowohl den Coder zur Umwandlung von Sprache in PCM als auch den Decoder, um aus der PCM wieder Sprache zu machen. Der Preis soll in den USA um 40 Dollar liegen.

Von Valvo ist das **Lehrsystem** Instructor 50 erhältlich. Es wird mit Handbuch und Einführungskurs geliefert, enthält den Prozessor 2650 (Signetics), 0,5 kByte RAM, ein 2-kByte-Monitorprogramm im ROM und ein Interface für Kassettenrecorder. Die gleiche Firma nahm auch den Ein-Chip-Mikrocomputer *8021* in ihr Lieferprogramm auf, der von Intel entwickelt wurde und aufgrund eines Lizenzabkommens nun auch von der Philips-Tochter Signetics gebaut wird.

Weniger als 3000 DM soll ein **Mikrocomputer mit 40 kByte** statischem RAM und dem 6800 kosten. Er heißt SWTP 6800/2 (Vertrieb: COI). Als Software wird ein Monitorprogramm im ROM einschließlich des Mini-Floppy-Steuerprogramms mitgeliefert, und als Verbindung zur Außenwelt dient eine serielle ASCII-Schnittstelle.

Dipl.-Ing. Hans Runge

Der Autor dieses Beitrags kommt von der professionellen EDV und beschäftigt sich schon seit Jahren auch mit Mikrocomputern. Wir möchten Ihnen seinen Kommentar zu der Qualität und Brauchbarkeit gängiger µC-Systeme nicht vorenthalten, nennt er doch die wesentlichsten Aspekte, die ein Anfänger beim Kauf seines ersten Mikrocomputers beachten sollte.

Welches Mikroprozessor-System für Anfänger

Kritisches zur Systemauswahl

Der Anfänger kauft meist ein zunächst preiswert erscheinendes System. Doch hinterher kommt oft der billigste „low cost"-Bausatz am teuersten...

An das Wichtigste denkt er oft zuletzt: Die für den ausgewählten Mikroprozessor-Typ erhältlichen Programme. Das Erwachen kommt, wenn er z. B. mit dem KIM-1 (Platine mit dem Mikroprozessor 6502) für das Zusammenzählen zweier 2stelliger Hexadezimal-Zahlen etwa 64 Tasten-Drücke braucht. Allerdings wird dieses Programm mitgeliefert und läßt sich auf einer normalen Tonkassette speichern.

Ohne entsprechende Programme läuft kein Mikroprozessor – noch weniger ein Klein-Computer, für den wenigstens immer einige System-Programme mitgeliefert werden, z. B. zum Umwandeln eines BASIC-Programmes in die Maschinen-Sprache, die beim Computer umständlich in kleinsten Schritten arbeitet.

Vom Aufwand und den verfügbaren Programmen her kommen praktisch nur wenige (8-bit-)Mikroprozessoren in Betracht. Diese sind:

- *8080* und *8085* von Intel (und etwa acht Firmen als Zweitlieferanten, auch von Siemens).
  8080-Programme laufen auch mit dem Z-80 (Z-80 A) von Zilog. Dieser Mikroprozessor hat alle 78 Befehle des 8080, darüber hinaus jedoch weitere 70 Befehle, die teuren Speicherplatz sparen. Dadurch gleicht sich der hohe Preis des Z-80 wieder aus *(Bild 1)*.

- *6502* von MOS Technology: Dieser wurde aus dem 6800 entwickelt und hat einen mit dem 6800 kompatiblen Bus (Daten- und Steuer-Leitungen) zum Anschalten von Peripherie-Geräten wie Drucker, Speicher etc. Durch überlappende Befehlsverarbeitung arbeitet der 6502 auch mit 1 MHz Taktfrequenz sehr schnell.

  Auch durch Verwendung im PET 2001, einem kompletten Computer unter 3000 DM, und im KIM 1, der weit verbreitet ist, dürften für den 6502 wenig Probleme in der Beschaffung von grundlegenden Programmen bestehen; ob allerdings so zahlreich wie beim 8080, sei dahingestellt.

- Der *6800* von Motorola ist der älteste der neuen Generation. Für ihn existieren viele ältere Programme. Auch ist Motorola als zuverlässiger Lieferant bekannt, was man sonst von den wenigsten der bekannten Hersteller in gleichem Maße behaupten kann.

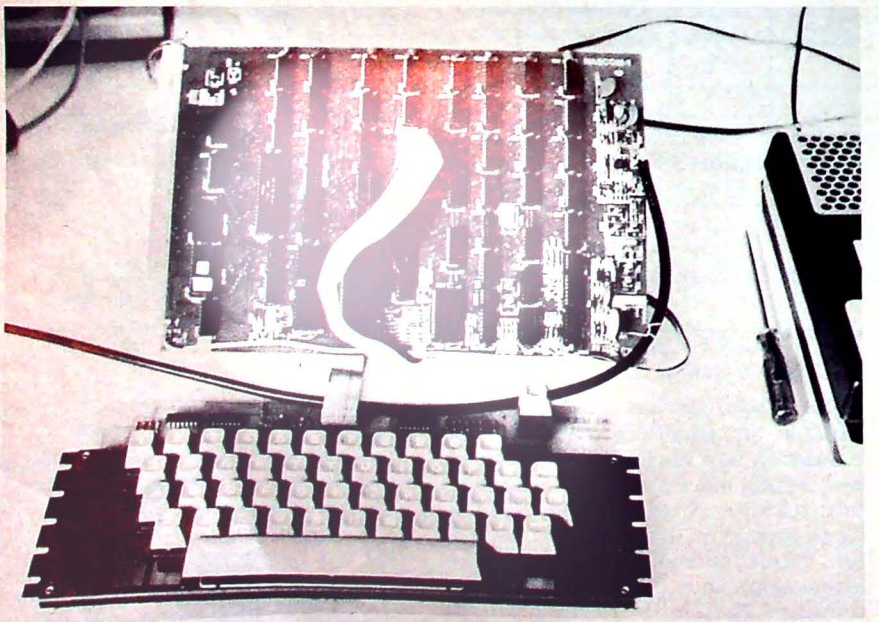


Bild 1. Video-Interface und ASCII-Tastatur rund um den Mikroprozessor Z 80: Der Mikrocomputer NASCOM. Nachteil: Die Schnittstellen sind zum Teil nicht genormt

Wichtig ist auch die anschließbare Hardware (Peripherie) um den Mikroprozessor herum: Am teuersten sind Drucker und Disketten-Laufwerke. Sie müssen alle irgendwie zusammenpassen, d. h. mit der Zentral-Einheit (CPU) bzw. der Mikroprozessor-Platine kompatibel sein. Der Anfänger kauft deshalb alles möglichst bei einer Quelle, die auch dieses Zusammenpassen garantiert.

Im fortgeschrittenen Stadium kann er dann selbst aus einem reichhaltigen Angebot auswählen, vorausgesetzt, er hat schon zu Anfang die richtige Auswahl getroffen. Der S-100-Bus ist recht verbreitet; er besteht aus einem Flachstecker mit 2 x 50 Kontakten, die Platine hat entsprechend 2 x 50 vergoldete Kontakte beidseitig. Er wurde erstmalig beim IMSAI/ALTAIR-Minicomputer verwendet. Die Anschlußbelegung mutet an, als ob ein Bastler sie schnell zusammengelötet hätte. Kritisiert wird

manchmal auch eine Vielzahl von z. T. entbehrlichen Steuerleitungen. Nichtsdestotrotz ist der S-100-Bus für Selbstbaugeräte und Heimcomputer der am meisten verwendete Bus, insbesondere in den USA. Daher sind auch Platinen für Speicher und andere Peripherie-Geräte für den S-100-Bus am reichhaltigsten vorhanden und durch die größeren Stückzahlen meist auch am preiswertesten. Der S-100-Bus ist jedoch sehr am 8080-Mikroprozessor (bzw. Z-80) orientiert.

Wie schon beim Computermarkt zeigt sich hier, daß Mikroprozessoren selbst bei gleicher oder überragender Qualität kaum wettbewerbsfähig sind, wenn sie nicht mit dem marktbeherrschenden System (hier dem 8080) voll kompatibel sind, wie einzig der Z-80 mit dem 8080 und dem S-100-Bus.

Auch nachdem die Auswahl des Anfängers vielleicht auf den 8080 oder Z-80 gefallen ist, sollte er wissen:

- Der 8080 braucht als Versorgungsspannung neben +5 Volt noch -5 V und +12 V; nur der Nachfolge-Typ 8080A und der 8085 kommen mit zwei positiven Spannungen, bzw. nur mit +5 V aus.

- Der Z-80 braucht nur +5 V und verwendet eine Taktfrequenz von 2 MHz. Der nur geringfügig teurere Z-80A läßt eine Taktfrequenz von 4 MHz zu und hat damit die doppelte Verarbeitungsgeschwindigkeit. Zur vollen Ausnützung müßten allerdings auch schnellere Speicher-Versionen verwendet werden, was dann etwa 20% teurer wird.

- Um möglichst mit einem 5-V-Netzteil auszukommen, sollte man anstelle der mit „komplizierter" Stromversorgung arbeitenden elektrisch programmierbaren Festwert-Speicher (EPROM) 2708, 2716 u. ä. nur 5-V-Typen wie 2758 verwenden. Auch bei den RAM-Speichern, deren Speicher-Inhalt nach dem Ausschalten verloren geht, sollte man möglichst große Einheiten verwenden: Sie verbrauchen weniger Strom, sind betriebssicherer und im Endeffekt kaum teurer, wenn auch z. T. schwerer zu haben.

Der Anfänger möchte erst einmal etwas lernen und zögert, gleich einen doch recht teuren Mini-Computer mit BASIC zu kaufen. Beim PET 2001 kann er zunächst nicht einmal in der Assembler-Sprache programmieren. Daher sollte die Auswahl möglichst auf ein Gerät fallen, das auch in Maschinensprache programmierbar ist und damit wesentlich lehrreicher und flexibler ist.

Leider wird der Anfänger oftmals durch Fachzeitschriften und Lehrinstitute auf einen recht kostspieligen Irrweg geführt: Vor den so häufig empfohlenen *SC/MP*-Mikroprozessor-Lehr-Bausätzen kann ich aus eigenem Leid nur warnen. Mir wurde so auch ein SC/MP-„low cost"-Bausatz (Introkit) angedreht, der auf den ersten Blick recht preiswert erschien. Der Lieferant scheint jedoch die „Qualität" treffend einzuschätzen. Denn er vermerkt auf der Rechnung: „Auf Bausätze kann weder von der Firma National Semiconductor noch von uns Garantie geleistet werden."

Mit diesem SC/MP-Introkit mit dem Mikroprozessor ISP-8A-500 läßt sich wenig anfangen, es sei denn, man unterzieht einen teuren Fernschreiber einem zweckentfremdenden mechanischen Umbau. Dazu wird die Gebrauchsanweisung mitgeliefert! Sonst enthält die Platine kaum etwas außer einem 20-mA-Serienanschluß für diesen Fernschreiber, einem nur dafür verwendbaren Programm in einem teuren ROM gespeichert und ganze 256 Byte in einem RAM als Programm-Speicher.

Man kann – zum gleichen Preis in etwa – noch eine Tastatur mit Anzeige dazukaufen (ein offenbar verschrotteter Billigst-Taschen-Rechner). Der auf der Anzeige-Platine in schwarzer Kunststoff-Vergußmasse eingebettete Rechen-Chip wurde schamhaft im Schaltbild unterschlagen. Dazu kommen noch statt der bei Bausätzen üblichen Print-Platine 125 himmelblaue Drähte einer Länge. Für eine spätere elektrische Fehlersuche fehlt leider eine Beschreibung.

Noch „eleganter" ist die Anschaltung an den Introkit: Der Mikroprozessor ISP-8A-500 braucht neben +5 V ungebräuchliche −7 V (Nachfolge-Typ ISP-8A-600: nur 5 V). National Semiconductor gewinnt diese −7 V aus 12 V: Beim Aussetzen der +5-V-Spannung (z. B. Drahtbruch) liegen die 12 V fast voll an der +5-V-Leitung!

Ich empfehle daher doch etwas anderes, nämlich zum ungefähr gleichen Gesamtpreis lieber gleich eine fertig zusammengebaute KIM-1-Platine mit dem 6502-Mikroprozessor und einem Kassetteninterface zu kaufen *(Bild 2)*. Sie enthält dies alles bereits zusammengebaut, dazu noch einiges mehr, z. B. 1024 Byte RAM-Speicher, also das Vierfache! Auch die KIM-1-Handbücher sind zwar didaktisch nicht gerade gelungen, aber selbst bei teureren Computern wird dem Anfänger in der Regel eine unmögliche Beschreibung zugemutet!

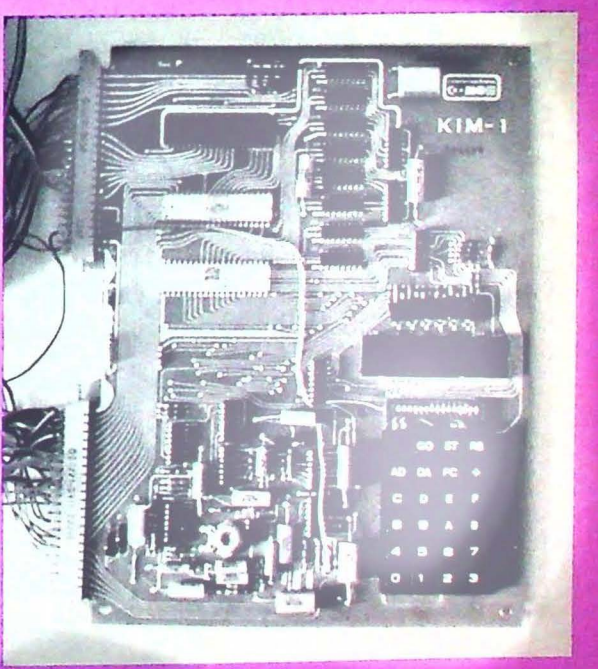
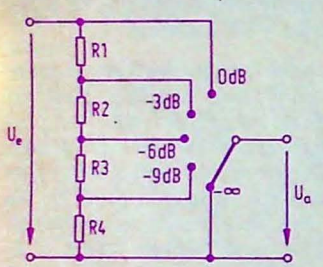


Bild 2. Der KIM-1, ein verbreitetes 6502-System mit 1,1-kByte-R und Kassetten-Interface (Fotos: Fe.)

Berechnung eines Eingangsteilers mit dem SR-56

Besonders in der Meßtechnik findet man häufig Eingangs-Spannungsteiler, die aus einer Kette von Widerständen aufgebaut sind und z. B. dazu dienen, eine dekadische Meßbereichsumschaltung bei Transistorvoltmetern zu ermöglichen. Das hier gezeigte Programm macht die Dimensionierung zum Kinderspiel.

Eingangsspannungsteiler sehen normalerweise so aus, wie es aus dem *Bild* hervorgeht: Der „L-Regler" besteht aus einem Stufenschalter mit z. B. fünf Stellungen. In jeder Schaltstellung läßt sich der Teiler auf einen einfachen



Das hier vorgestellte Programm ermöglicht die rasche Berechnung der Widerstände dieser Teilerkette für bestimmte Dämpfung pro Schaltschritt, hier 3 dB

Spannungsteiler aus nur zwei Widerständen zurückführen, indem man die oberhalb und unterhalb des Anzapfpunktes liegenden Widerstände jeweils summiert. Das Programm für den SR-56 (*Tabelle*); es gestattet die Eingabe der Dämpfung pro Schaltschritt in dB – eine Tatsache, die besonders die Berechnung von stufenweise einstellbaren Lautstärkereglern in komfortablen Hi-Fi-Verstärkern o. ä. erleichtert. Zu beachten ist, daß Voraussetzung für ein relevantes Ergebnis ein leerlaufender Spannungsteiler ist, d. h. der Ausgang des Stufenschalters muß mit einem Impedanzwandler verbunden sein, dessen Eingangswiderstand deutlich größer als die Summe aller Widerstände der Teilerkette ist.

Selbstverständlich läßt sich das Programm leicht auch für andere Rechnertypen, z. B. TI-58/59, modifizieren; die Befehle „Pause" können dann bei Vorhandensein eines Druckers in „Print" geändert werden.

Die Bedienung des Programms erfolgt nach dem Eintippen der Befehle wie folgt:

Eingabe	Taste
Gesamtwiderstand R_g	2nd CMS RST STO 7
Schrittdämpfung/dB	STO 2
Gesamtdämpfung $^1/_4$s anzeigen und R 1 berechnen	R/S
Verbleibender Restwiderstand	R/S
Weitere Widerstände berechnen	R/S
Änderung der Schrittdämpfung/dB	STO 2
Widerstände berechnen	R/S

Wie zu erkennen ist, läßt sich die Dämpfung pro Schaltschritt nach dem Programmdurchlauf ohne weiteres ändern. Dann wird das Programm erneut mit R/S gestartet. *Winfried Neumann*

Tabelle: Programm für den SR-56

Code	Taste	Code	Taste
56	2nd CP	32	↔
34	RCL	94	=
02	2	49	2nd Fix
35	SUM	02	2
03	3	33	STO
49	2nd Fix	04	4
01	1	32	↔
12	INV	74	–
44	EE	34	RCL
34	RCL	01	1
03	3	74	–
59	2nd Pause	32	↔
54	:	94	=
02	2	35	SUM
00	0	01	1
94	=	41	R/S
19	2nd 10^x	34	RCL
32	↔	04	4
34	RCL	41	R/S
07	7	42	RST
54	:		

Der Heath-Computer

Die Firma Heathkit, unseren amateurfunkenden Lesern schon länger durch ihre Transceiverbausätze und anderes bekannt, steigt nun auch in das Hobbycomputer-Geschäft ein. Das von Heath angebotene System umfaßt den modular aufgebauten H-8-Computer mit dem Mikroprozessor 8080 sowie hexadezimaler Eingabe und Anzeige, einem Video-Terminal mit 31-cm-Schirm (allerdings nur mit Großschreibung), einen Lochstreifen-Stanzer und -Leser, ein Floppy-Disk-Laufwerk und andere periphere Geräte.

Für das System steht eine umfangreiche Software zur Verfügung: Benton Harbor BASIC, Assembler, Text-Editor und Debugging-Programm. Sie wird auf Magnetkassetten geliefert.

Für die Verbindung der Bausteine untereinander wird ein spezieller Bus verwendet, der nicht mit anderen kompatibel ist, aber wegen seiner vollständigen Bufferung ohne weiteres Expansionen zuläßt.

Die Anzeige von Adressen und Daten erfolgt mit dem H-8-Computer nicht hexadezimal, sondern oktal – vielleicht ewas ungewohnt. Ein Interface zum Anschluß eines Kassettenrecorders als preiswerter Programmspeicher ist bereits eingebaut. Allerdings ist der Spaß nicht ganz billig: Ein System mit dem Computer, dem Videoterminal und einem Kassettenrecorder kostet rund 4400 DM.

Unser *Bild* zeigt das Heath-Terminal, den Computer H 8 und den Lochstreifenleser bzw. -stanzer.

Das Heathkit-Computersystem kann als Bausatz oder auch in zusammengebauter Form bezogen werden

Auf der Hobbyelektronik-Ausstellung in Wien war auf dem FUNKSCHAU-Stand eine recht interessante Mikrocomputer-Anwendung zu sehen, die die Möglichkeiten einer Textverarbeitung demonstrieren sollte, nämlich ein Programm zur automatischen Formatierung von gespeicherten Texten mit beliebiger Zeilenlänge. Hier ist zu lesen, wie's funktioniert.

Automatische Text-Formatierung

Wenn Sie diesen Beitrag lesen, machen Sie sich wahrscheinlich kaum Gedanken darüber, warum alle Zeilen einer Textspalte immer genau gleich lang sind und welcher technische Aufwand dahinter steckt, diese Formatierung und auch die richtige Trennung von Worten zu erreichen. Früher war dies Aufgabe der Schriftsetzer; heute wird die FUNKSCHAU mit einem Computer gesetzt, dem ein fortlaufender Text (also ohne Zeilenwechsel- oder Trennungszeichen) eingetippt wird. Der Computer sorgt dann von selbst dafür, daß genau die gewünschte Zeilenlänge eingehalten wird, und dank eines umfangreichen Programms ist er sogar schlau genug, Worte an der richtigen Stelle zu trennen.

Nun, ganz so weit wollen wir hier nicht gehen, sonst bleibt bei den üblichen Mikrocomputer-Systemen ja kein Speicherplatz mehr für den Text selbst übrig. Zumindest erlaubt das in *Bild 1* gezeigte Flußdiagramm aber die richtige Trennung am Ende einer Zeile, nämlich genau an der Stelle, wo ein Leerraum zwischen zwei Worten steht (die Worte selbst werden also grundsätzlich nicht getrennt); und nicht zuletzt sorgt es durch gezieltes Einfügen von zusätzlichen Leerräumen innerhalb einer Zeile dafür, daß die gewünschte Zeilenlänge exakt erreicht wird. Mit anderen Worten: Das Programm besorgt die formatierte Ausgabe eines gespeicherten Textes mit Randausgleich.

Bild 2 zeigt schließlich, daß der Aufwand für eine solche formatierte Textausgabe gar nicht sehr groß ist. Das aufgelistete Programm wurde für den Mikrocomputer KIM-1 geschrieben und geht davon aus, daß der auszudruckende Text in Form beliebiger ASCII-Zeichen an der Adresse 0200 beginnt. Die Zeilenlänge kann durch Einschreiben des entsprechenden hexadezimalen Wertes (max. FF = 255) in die Zelle 00EE frei gewählt werden.

Die Ausgabe wird beendet, sobald das Programm im Text auf den Wert 00 stößt; dann erfolgt ein Sprung zum KIM-Monitorprogramm an die Adresse 1C4F. Für alle, die ein anderes System benutzen, sei noch erwähnt, welche Bedeutung die vorkommenden Unterprogramm-Sprünge in das Monitorprogramm haben:

Unterprogramm	Adresse
Ausdruck eines Leerraums	1E9E
Ausdruck eines Zeichens	1EA0
Erhöhen des Zeigers (FA, FB)	1F63

Zum Schluß sei noch erwähnt, daß das Programm noch einige Variationsmöglichkeiten bietet. Z.B. wäre es ohne weiteres möglich, bei Erkennen eines Wagenrücklaufzeichens im gespeicherten Text die aktuelle Zeile gewaltsam zu beenden und eine neue zu beginnen, um die Möglichkeit zu schaffen, Absätze in den Text einzufügen. Wie bei allen Programmvorschlägen, so ist auch hier das Können des Computers in erster Linie durch den Zeitaufwand begrenzt, den man in die notwendigen Programme investiert!

Herwig Feichtinger

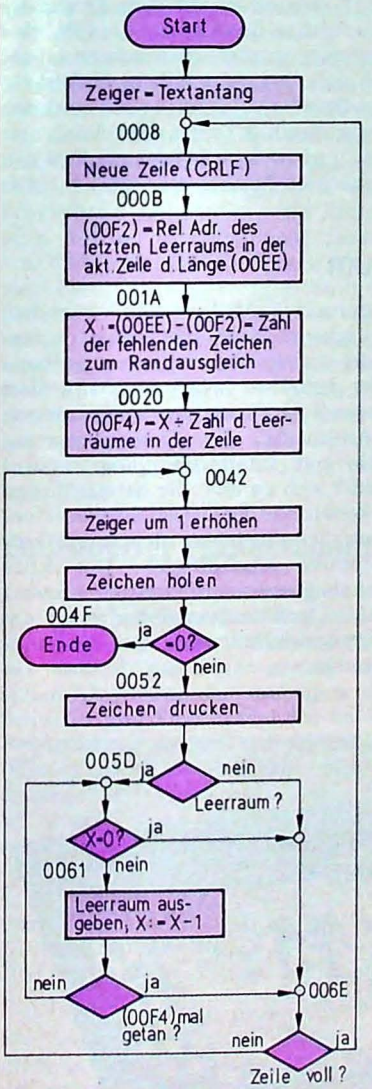


Bild 1. Flußdiagramm des Formatierungs-Programms

```
0200 A9 00   LDA #$00      0010 E5 F2   SBC $F2      0025 D0 F5   BNE 001C      0052 85 F9   STA $F9
0202 85 FA   STA $FA       001F AA      TAX          0027 E4 F3   CPX $F3      0054 20 9E 1E JSR $1E9E
0204 A9 02   LDA #$02      0020 A9 00   LDA #$00     0029 D0 04   BNE 002F      0057 A5 F9   LDA $F9
0206 85 FB   STA $FB       0022 E5 F3   STA $F3      002B E6 F4   INC $F4      0059 C9 20   CMP #$20
0208 20 2F 1E JSR $1E2F    0024 85 F4   STA $F4      002D D0 E4   BNE 0013      005B D0 11   BNE 006E
020B A4 EE   LDY $EE       0026 EA      NOP          002F EA      NOP           005D E0 00   CPX #$00
020D B1 FA   LDA ($FA),Y   0027 EA      NOP          0030 EA      NOP           005F F0 06   BEQ 0067
020F C9 20   CMP #$20      0028 EA      NOP          0031 EA      NOP           0061 20 9E 1E JSR $1E9E
0211 F0 05   BEQ 0018      0029 A4 F2   LDY $F2       0032 A5 F4   LDA $F4      0064 CA      DEX
0213 88      DEY           002B 88      DEY          0034 85 F3   STA $F3      0065 A5 F3   LDA $F3
0214 D0 F7   BNE 000D      002C B1 FA   LDA ($FA)    0036 20 63 1F JSR $1F63    0067 F0 05   BEQ 006E
0216 A4 EE   LDY $EE       002E C9 20   CMP #$20     0039 A0 00   LDY #$00      0069 C6 F2   DEC $F3
0218 84 F2   STY $F2       0030 D0 02   BNE 0034     003B B1 FA   LDA ($FA),    006B 18      CLC
021A A5 EE   LDA $EE       0032 E6 F2   INC $F3      003D D0 D3   BNE 0012      006C 90 EF   BCC 005D
021C 38      SEC           0034 88      DEY          003F 4C 4F 1C JMP $1C4F    006E C6 F2   DEC $F2
                                                                               0070 D0 D9   BNE 004C
                                                                               0072 4C 00 00 JMP $0000
```

Bild 2. Programm zum formatierten Ausdrucken eines gespeicherten Textes ab der Adresse 0200 für den Mikrocomputer KIM-1 (CPU: 6502)

KIM-Debugger

Mit wenig Aufwand ist es möglich, einen recht komfortablen Einzelschritt-Betrieb oder Breakpoint-Routinen auf dem KIM-1 zu realisieren. Auf einem angeschlossenen Terminal oder Fernschreiber wird dabei der Inhalt aller internen CPU-Register ausgedruckt, und zwar in folgendem Format:

```
X  Y  A  SP NV-BDIZC
01 02 03 FF 0110 0010
KIM
0204 A9
```

Die angegebenen Zahlen mögen hier nur als Beispiel dienen; es ist aber zu erkennen, daß das Statusregister nicht in Form zweier hexadezimaler Ziffern, sondern wirklich binär ausgegeben wird, so daß sofort der Zustand jedes einzelnen Bit zu sehen ist. Die Titelzeile (X, Y, A...) wird ebenfalls ausgedruckt.

```
START
(178o)
85 f3    Akku retten
68
85 f1    Statusreg.
68        retten
85 fa    Programm-
68        zähler
85 fb     retten
84 f4    Y-Index u.
86 f5    X-Index
ba
86 f2    Stackpointer
2o 88 1e I/O initial.
2o 2f 1e Neue Zeile
a2 14
bd cc 17 Überschrift
2o ao 1e  ausgeben
ca
1o f7
2o 2f 1e Neue Zeile
a2 o4
b5 f1     X,Y,A,SP
2o 3b 1e   ausgeben
2o 9e 1e
ca
do f5
a2 o8     Statusreg.
a5 f1      binär aus-
85 f9      geben
o6 f9
bo o4
a9 3o
9o o2
a9 31
2o ao 1e
ca
do fo
4c 4f 1c  Spr.z.KIM
43 5a 49  ASCII-
44 42 2d   Tabelle
56 4e 2o   für die
5o 53 2o   Überschrift
2o 41 2o
2o 59 2o
2o 58
END (17eo)
```

Bild 1. Auflistung des Debugging-Programms in der 6502-Maschinensprache

Das für diesen Ausdruck erforderliche kleine Programm findet leicht in dem etwas abseits liegenden RAM-Bereich des KIM-1 (Adressen ab 1780) Platz, so daß der normalerweise für Anwenderprogramme benutzte Raum (0000...03FF) dadurch nicht eingeschränkt wird.

Nun muß allerdings dafür gesorgt werden, daß das in *Bild 1* dargestellte Programm nicht selbst auch im Einzelschritt-Betrieb durchlaufen wird, denn dies soll ja nur im Anwenderprogramm geschehen. Wenn also der für den Einzelschritt maßgebende NMI-Vektor (Adressen 17FA und 17FB) auf die Programm-Startadresse 1780 zeigt, so soll das Register-Ausdrucken nicht im Single-Step-Modus erfolgen. Dazu ist nur eine winzige Hardware-Änderung erforderlich *(Bild 2)*. Ein in Serie mit dem Single-Step-Schalter am KIM-1 liegender Transistor verhindert diese Betriebsart, solange der Adressenbereich des RAM ab 1780 angesprochen wird. Das dazu erforderliche Signal wird vom Adressen-Decoder auf der KIM-Platine gewonnen.

Bei längeren Anwenderprogrammen ist der Einzelschritt-Modus etwas zeitaufwendig. Es ist dann günstiger, sogenannte „Breakpoints" zu setzen, die beim 6502 dem Hex-Code 00 entsprechen und an die Adresse geschrieben

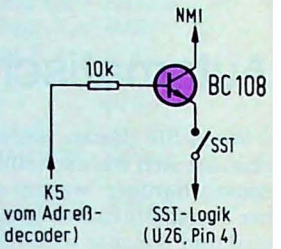


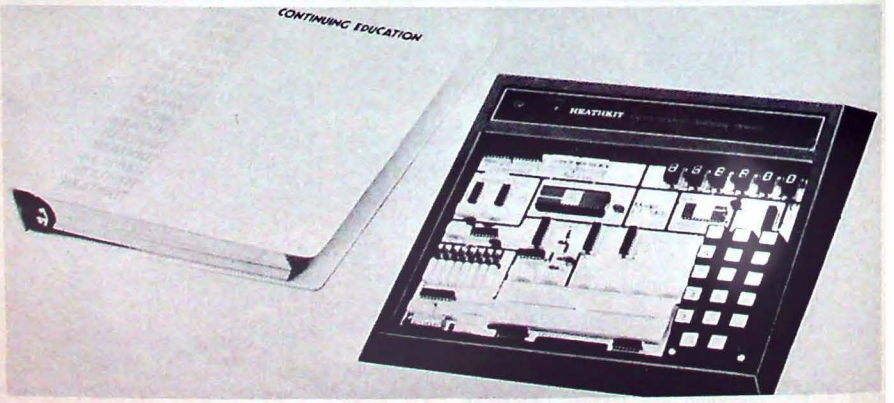
Bild 2. Notwendige Hardware-Änderung am KIM, um einen Single-Step-Ablauf des Programms ab der Adresse 1780 zu verhindern

werden, wo das Programm anhalten soll. Schreibt man in die Zellen 17EE und 17FF als IRQ-Vektor die Daten 80 und 17, so werden auch beim Erreichen eines solchen Breakpoints die CPU-Register ausgedruckt. Zu beachten ist jedoch, daß die CPU den Break-Befehl als Zwei-Byte-Befehl interpretiert und der Programmzähler (unter dem Ausdruck „KIM") nicht die nächste, sondern die übernächste Adresse anzeigt. Fe.

6800-System zum Selbststudium

Heathkit brachte einen kleinen Mikrocomputer für etwa 600 DM heraus, der mit der CPU 6800 arbeitet und über ein ROM mit 1 kByte Monitorprogramm verfügt. Auf der Platine sind auch 256 Byte RAM untergebracht, die ohne zusätzlichen Decoder auf 512 Byte erweitert werden können; ein 40poliger Verbindungsstecker gestattet den Anschluß zusätzlicher Speicher oder Geräte. Die Ein- und Ausgabe erfolgen mit einer Hexadezimal-Tastatur und einem sechsstelligen Siebensegment-Display. Die Adressen-, Daten- und Steuerleitungen sind bereits gebuffert. Das System wird in Bausatzform geliefert.

Ein rund die Hälfte davon kostender, englischsprachiger, 850 Seiten umfassender und auf diesem System aufbauender Lehrkurs befaßt sich mit den notwendigen Grundlagen, der Computer-Arithmetik, der Programmierung sowie mit Interface-Problemen und wendet sich an alle, die bereits einige Vorkenntnisse der Digitalelektronik besitzen, sich aber bisher nicht mit Mikrocomputern beschäftigten. Der Kurs wird zusammen mit 62 Bauelementen (RAM, Operationsverstärker u. a.) geliefert und enthält auch zwei Tonbandkassetten.



System fürs Selbststudium: ET-3400 von Heathkit